

## Esercizi di XML

Ombretta Gaggi  
Università di Padova

### Esercizio 1

- Descrivere uno schema per un elemento che descrive stringhe binarie
- binario contiene una serie di elementi uno o zero in qualsiasi ordine e con qualsiasi molteplicità

```
<binario>
  <uno>1</uno>
  <zero>0</zero>
  <uno>1</uno>
  <uno>1</uno>
  <uno>1</uno>
  <zero>0</zero>
</binario>
```

Tecnologie Web - 2

### Soluzione DTD

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

Tecnologie Web - 3

### Soluzione XML Schema

```
<xsd:element name="binario">
  <xsd:complexType>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="zero" type="xsd:unsignedByte" />
      <xsd:element name="uno" type="xsd:unsignedByte" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Tecnologie Web - 4

### Documenti istanza validi

<code>&lt;binario&gt;</code>	<code>&lt;binario&gt;</code>
<code>  &lt;uno&gt;1&lt;/uno&gt;</code>	<code>  &lt;uno&gt;0&lt;/uno&gt;</code>
<code>  &lt;zero&gt;0&lt;/zero&gt;</code>	<code>  &lt;zero&gt;1&lt;/zero&gt;</code>
<code>  &lt;uno&gt;1&lt;/uno&gt;</code>	<code>  &lt;uno&gt;0&lt;/uno&gt;</code>
<code>  &lt;uno&gt;1&lt;/uno&gt;</code>	<code>  &lt;uno&gt;1&lt;/uno&gt;</code>
<code>  &lt;uno&gt;1&lt;/uno&gt;</code>	<code>  &lt;uno&gt;0&lt;/uno&gt;</code>
<code>  &lt;zero&gt;0&lt;/zero&gt;</code>	<code>  &lt;zero&gt;0&lt;/zero&gt;</code>
<code>&lt;/binario&gt;</code>	<code>&lt;/binario&gt;</code>

Tecnologie Web - 5

### Documenti istanza validi (per DTD)

```
<binario>
  <uno>a</uno>
  <zero>0</zero>
  <uno>8</uno>
  <uno>5</uno>
  <uno>t</uno>
  <zero>0</zero>
</binario>
```

Tecnologie Web - 6

## Soluzione XML Schema - raffinamento

```
<xsd:element name="binario">
  <xsd:complexType>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="zero" type="xsd:unsignedByte"
        fixed="0" />
      <xsd:element name="uno" type="xsd:unsignedByte"
        fixed="1" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Tecnologie Web - 7

## Documenti istanza validi

<binario>	<binario>
<uno>1</uno>	<uno>1</uno>
<zero>0</zero>	<zero>/>
<uno>1</uno>	<uno>/>
<uno>1</uno>	<uno>/>
<uno>1</uno>	<uno>/>
<zero>0</zero>	<zero>0</zero>
</binario>	</binario>

Tecnologie Web - 8

## Esercizio 2

- Definire uno schema che permetta di validare il seguente documento XML, utilizzando il modello Giardino dell'Eden:

```
<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>
    <articolo><codice>5</codice>
    <descr>articolo 5</descr></articolo>
  </ordine>
  Cordiali saluti,
  <responsabile><tit>dr. </tit> Mario Rossi</responsabile>
</lettera>
```

Tecnologie Web - 9

## 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"/>
  <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>
```

Tecnologie Web - 10

## Esercizio 2 - soluzione - 2

```
<xsd:complexType name="Tordine" >
  <xsd:sequence>
    <xsd:element ref="articolo" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute ref="num" />
</xsd:complexType>
<xsd:complexType name="Tarticolo"><xsd:sequence>
  <xsd:element ref="codice" /><xsd:element ref="descr" />
</xsd:sequence></xsd:complexType>
<xsd:complexType name="Tresponsabile" mixed="true" >
  <xsd:choice minOccurs="0">
    <xsd:element ref="tit" />
  </xsd:choice>
</xsd:complexType>
```

Tecnologie Web - 11

## 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"
        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="num" type="xsd:integer"/>
  </xsd:complexType>
```

Tecnologie Web - 12

## Esercizio 2 - soluzione tende alla veneziana - 2

```
<xsd:complexType name="Tarticolo">
  <xsd:sequence>
    <xsd:element name="codice" type="xsd:string"/>
    <xsd:element name="descr" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Tresponsabile" mixed="true">
  <xsd:choice minOccurs="0">
    <xsd:element name="tit" type="xsd:string"/>
  </xsd:choice>
</xsd:complexType>
```

Tecnologie Web - 13



## 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>

    <articolo>articolo 5</articolo>
  </ordine>
  Cordiali saluti,
  <responsabile><tit>dr.</tit> Mario Rossi</responsabile>
</lettera>
```

Tecnologie Web - 14



## Soluzione

```
<xsd:complexType name="Tordine">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="articolo" type="xsd:string"/>
    <xsd:element name="articolo">
      <xsd:complexType><xsd:sequence>
        <xsd:element ref="codice"/>
        <xsd:element ref="descr"/>
      </xsd:sequence></xsd:complexType>
    </xsd:element>
  </xsd:choice>
  <xsd:attribute ref="num"/>
</xsd:complexType>
```

Tecnologie Web - 15



## Soluzione corretta

```
<xsd:complexType name="Tarticolo" mixed="true">
  <xsd:sequence minOccurs="0">
    <xsd:element ref="codice"/>
    <xsd:element ref="descr"/>
  </xsd:sequence>
</xsd:complexType>

□ Però viene validato anche
<articolo>articolo 5
  <codice> 5 </codice> <descr> articolo 5 </descr>
</articolo>
```

Tecnologie Web - 16



## 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>
</ordine>

<lettera xmlns=...>
  Gentile <cliente>Tal dei Tali</cliente>,
  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
    più 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>
```

Tecnologie Web - 18



## Esercizio 3

- Definire uno schema che permetta di validare il seguente documento:
 

```
<montagne>
  <catenaMontuosa>
    <monte>
      <nome>Monte Bianco</nome>
      <regione>Valle d'Aosta</regione>
      <altezza unitaMisura="metri">4810</altezza>
    </monte>
  </catenaMontuosa>
  <catenaMontuosa>
    <monte><nome>Gransasso</nome> ... </monte>
  ...
</catenaMontuosa>
</montagne>
```

Tecnologie Web - 19



## 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

```
<xsd:schema xmlns... >
<xsd:element name="montagne">
  <xsd:complexType><xsd:sequence>
    <xsd:element ref="catenaMontuosa" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence> <xsd:attribute name="nome" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="catenaMontuosa">
  <xsd:complexType><xsd:sequence>
    <xsd:element ref="monte" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence></xsd:complexType>
</xsd:element>
```

Tecnologie Web - 21



## Soluzione XML Schema - 2

```
<xsd:element name="monte" >
  <xsd:complexType><xsd:sequence>
    <xsd:element ref="nome" />
    <xsd:element ref="regione" />
    <xsd:element ref="altezza" />
  </xsd:sequence></xsd:complexType>
</xsd:element>
```

Tecnologie Web - 22



## Soluzione XML Schema - 3

```
<xsd:element name="monte" >
  <xsd:complexType><xsd:sequence>
    <xsd:element ref="nome" />
    <xsd:element ref="regione" minOccurs="0" />
    <xsd:element ref="altezza" minOccurs="0" />
  </xsd:sequence></xsd:complexType>
</xsd:element>
```

Tecnologie Web - 23



## 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:element>
```

Modello Fette di Salame

Tecnologie Web - 24



### Soluzione Modello Bambole Russe - 1

```
<xsd:element name="montagne" ... >
  <xsd:element name="catenaMontuosa" minOccurs="0" ...>
    <xsd:complexType><xsd:sequence>
      <xsd:element name="monte" maxOccurs="unbounded"/>
      <xsd:complexType><xsd:sequence>
        <xsd:element name="nome" type="xsd:string"/>
        <xsd:element name="regione" type="xsd:string"/>
        <xsd:element name="altezza"/>
        ...
      </xsd:sequence></xsd:complexType>
    </xsd:sequence></xsd:complexType>
  </xsd:element>
  <xsd:attribute name="nome" type="xsd:string"/>
</xsd:element>
</xsd:element>
```

Tecnologie Web - 25

### Soluzione Modello Bambole Russe - 2

```
...
  <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:element>
```

Tecnologie Web - 26

### 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.

Tecnologie Web - 27

### Soluzione Esercizio 3 - raffinamento

```
<xsd:element name="montagne" ... >
  <xsd:element name="catenaMontuosa" maxOccurs="unbounded">
    <xsd:complexType><xsd:sequence>
      <xsd:element name="monte" maxOccurs="unbounded"/>
      <xsd:complexType><xsd:sequence>
        <xsd:element name="nome" type="xsd:string"/>
        <xsd:element name="regione" type="xsd:string"/>
        <xsd:element name="altezza"/>
        ...
      </xsd:sequence>
      <xsd:attribute name="nome" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:any minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence></xsd:complexType>
</xsd:element>
</xsd:element>
```

Tecnologie Web - 28

### Finito?

- In generale sì, ma io posso raffinare ulteriormente il mio schema:

```
<xsd:element name="regione" >
  <xsd:simpleType >
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Valle d'Aosta"/>
      <xsd:enumeration value="Piemonte"/>
      <xsd:enumeration value="Lombardia"/>
      ...
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Tecnologie Web - 29

### Conversione in una pagina web

- Definire una trasformata XSLT che realizzi la seguente pagina web

Tecnologie Web - 30

## Il file XML

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="montagne.xml"?>
<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>
</montagne>
```

Tecnologie Web - 31



## La trasformata - 1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8"
    indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Montagne Italiane</title>
        <style type="text/css" media="all">
          .attenzione{ color:red;font-variant: small-caps;}
        </style>
      </head>
```

Tecnologie Web - 32



## La trasformata - 2

```
<body>
  <h1>Le montagne italiane</h1>
  <p> In Italia sono presenti le seguenti catene montuose: </p>
  <ul>
    <xsl:for-each select="montagne/catenaMontuosa">
      <li> <xsl:value-of select="@nome"/></li>
    </xsl:for-each>
  </ul>
  <p>Il nostro sito vi propone di visitare i monti pi&#250; interessanti al loro interno, in particolare: </p>
  <ul><xsl:for-each select="montagne/catenaMontuosa/monte">
    ...</xsl:for-each>
  </ul>
</body> </html> </xsl:template> </xsl:stylesheet>
```

Tecnologie Web - 33



## La trasformata - 3

```
<p>Il nostro sito vi propone di visitare i monti pi&#250; interessanti al loro interno, in particolare: </p>
<ul><xsl:for-each select="montagne/catenaMontuosa/monte">
  <xsl:sort select="nome"/>
  <li xml:space="preserve"> <xsl:value-of select="nome"/>,
    situato in <xsl:value-of select="regione"/>,
    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>
  </li>
</xsl:for-each>
</ul>
```

Tecnologie Web - 34



## Namespace - il file XML

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="montagne.xml"?>
<montagne xmlns="http://www.montagne.it">
  <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>
</montagne>
```

Tecnologie Web - 35



## Namespace - la trasformata - 1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:m="http://www.montagne.it"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8"
    indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Montagne Italiane</title>
        <style type="text/css" media="all">
          .attenzione{ color:red;font-variant: small-caps;}
        </style>
      </head>
```

Tecnologie Web - 36



### Namespace - la trasformata - 2

```
<body>
<h1>Le montagne italiane</h1>
<p> In Italia sono presenti le seguenti catene montuose: </p>
<ul>
  <xsl:for-each select="m:montagne/m:catenaMontuosa">
    <li> <xsl:value-of select="@nome"/></li>
  </xsl:for-each>
</ul>
<p>Il nostro sito vi propone di visitare i monti pi&#250; interessanti al loro interno, in particolare: </p>
<ul><xsl:for-each select="m:montagne/m:catenaMontuosa/m:monte">
  ...</xsl:for-each>
</ul>
</body> </html> </xsl:template> </xsl:stylesheet>
```

Tecnologie Web - 37



### Namespace - la trasformata - 3

```
<p>Il nostro sito vi propone di visitare i monti pi&#250; interessanti al loro interno, in particolare: </p>
<ul><xsl:for-each select="m:montagne/m:catenaMontuosa/m:monte">
  <xsl:sort select="m:nome"/>
  <li xml:space="preserve"> <xsl:value-of select="m:nome"/>,
    situato in <xsl:value-of select="m:regione"/>,
    ha un'altezza pari a <xsl:value-of select="m:altezza"/>
    <xsl:value-of select="m:altezza/@unitaMisura"/>
    <xsl:if test="altezza &gt; 4000">
      <span class="attenzione"> (Solo per esperti) </span>
    </xsl:if>
  </li>
</xsl:for-each>
</ul>
```

Tecnologie Web - 38



### 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>
    <cell>353134125</cell>
    <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">50</prezzo>
    <prezzo stanza="singola">40</prezzo>
  </albergo>
</alberghi>
```

Tecnologie Web - 39



### 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 prezzo (#PCDATA) >
<!ATTLIST prezzo stanza CDATA #REQUIRED>
```

Tecnologie Web - 40



### Soluzione DTD: raffinamento

```
<!ATTLIST prezzo stanza (singola | doppia) #REQUIRED>
<!ATTLIST albergo categoria (* | ** | *** | ...) #REQUIRED>
```

NON CORRETTA!

Tecnologie Web - 41



### Ripasso: per associare una DTD ad un file XML

```
<?xml version="1.0"?>
<!DOCTYPE alberghi [
  <!ELEMENT alberghi (albergo)+>
  <!ATTLIST albergo categoria CDATA #REQUIRED>
  ...]>
<alberghi> ... </alberghi>

<?xml version="1.0"?>
<!DOCTYPE alberghi SYSTEM "alberghi.dtd">
<alberghi> ... </alberghi>
```

Tecnologie Web - 42



### Soluzione XMLSchema: namespace

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.alberghi.com"
  targetNamespace="http://www.alberghi.com"
  elementFormDefault="qualified">
  <xs:element name="alberghi">
    <xs:complexType><xs:sequence>
      <xs:element name="albergo" maxOccurs="unbounded">
        ...
      </xs:element>
    </xs:sequence></xs:complexType>
  </xs:element>
</xs:schema>
```

Tecnologie Web - 43



### 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="tel" type="xs:string"/>
    <xs:element name="cell" type="xs:string" minOccurs="0"/>
    <xs:element name="prezzo" maxOccurs="unbounded">
      ...
    </xs:element>
  </xs:sequence>
  <xs:attribute name="categoria" ... </xs:attribute>
</xs:complexType>
</xs:element>
```

Tecnologie Web - 44



### Soluzione XMLSchema: attributo categoria

```
<xs:attribute name="categoria">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="**"/>
      <xs:enumeration value="**"/>
      <xs:enumeration value="**"/>
      <xs:enumeration value="**"/>
      <xs:enumeration value="**"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

Tecnologie Web - 45



### Soluzione XMLSchema: elemento prezzo

```
<xs:element name="prezzo" maxOccurs="unbounded">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:decimal">
        <xs:attribute name="stanza" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Tecnologie Web - 46



### Foglio di trasformazione

#### 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

Tecnologie Web - 47



### Codice - 1

```
<xsl:template match="/">
  <html><head><title>Agenzia Turismo Italia</title>
    <style type="text/css" media="all">
      dt{font-variant: small-caps;}
    </style>
  </head>
  <body>
    <h1>Le nostre offerte</h1>
    ...
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Tecnologie Web - 48





## Codice - 2

```
<dl><xsl:for-each select="//albergo">
  <xsl:sort select="citta"/>
  <dt> <xsl:value-of select="nome"/>, <xsl:value-of select="citta"/>
    (telefono <xsl:value-of select="tel"/>) </dt>
  <dd> categoria: <xsl:value-of select="@categoria"/>
    <xsl:for-each select="prezzo">
      stanza <xsl:value-of select="@stanza"/>:
      <xsl:value-of select="."/>
    </xsl:for-each>
  </dd>
</xsl:for-each> </dl>
```

Tecnologie Web - 49



## Codice ricorsivo

```
<xsl:template match="/">
  ...<xsl:apply-templates select="//a:albergo">
    <xsl:sort select="a:citta"/>
  </xsl:apply-templates> ...
</xsl:template>
<xsl:template match="albergo" >
  <dt> <xsl:value-of select="nome"/>, <xsl:value-of select="citta"/>
    (telefono <xsl:value-of select="tel"/>) </dt>
  <dd> categoria: <xsl:value-of select="@categoria"/>
    <xsl:apply-templates />
  </dd>
</xsl:template>
<xsl:template match="prezzo">
  stanza <xsl:value-of select="@stanza"/>: <xsl:value-of select="."/>
</xsl:template>
```

Tecnologie Web - 50



## Altra trasformata

### Le nostre offerte

Queste sono le nostre offerte per alberghi a Milano:

PENSIONE UNA STELLA (TELEFONO 028872382)  
 categoria: \* stanza doppia: 40 stanza singola: 30  
 ALBERGO CARLOTI (TELEFONO 028872382)  
 categoria: \*\*\* stanza doppia: 160 stanza singola: 100

Queste sono le nostre offerte per alberghi a Garda:

ALBERGO MIRALAGO (TELEFONO 0458872382)  
 categoria: \*\*\*\*\* stanza doppia: 160 stanza singola: 100

Queste sono le nostre offerte per alberghi a Roma:

HOTEL ASTORIA (TELEFONO 06237813)  
 categoria: \*\*\*\*\* stanza singola: 120 stanza doppia: 200  
 PENSIONE SCATAFASCIO (TELEFONO 068872382)  
 categoria: \* stanza doppia: 50 stanza singola: 40

Tecnologie Web - 51



## Codice - 1

```
<xsl:template match="/">
  <html><head><title>Agenzia Turismo Italia</title>
  <style type="text/css" media="all">
    dt{font-variant: small-caps;}
  </style>
</head>
<body>
  <h1>Le nostre offerte</h1>
  ...
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Tecnologie Web - 52



## Codice - 2

```
<p>Queste sono le nostre offerte per alberghi a Milano: </p>
<dl><xsl:for-each select="//albergo[citta='Milano']">
  <dt> <xsl:value-of select="nome"/>
    (telefono <xsl:value-of select="tel"/>) </dt>
  <dd> categoria: <xsl:value-of select="@categoria"/>
    <xsl:for-each select="prezzo">
      stanza <xsl:value-of select="@stanza"/>:
      <xsl:value-of select="."/>
    </xsl:for-each>
  </dd>
</xsl:for-each> </dl>
```

Tecnologie Web - 53



## 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)">
    <p>Queste sono le nostre offerte per alberghi a <xsl:value-of
      select="a:citta"/> : </p>
  </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>
```

Tecnologie Web - 54



### Esercizio 5

```
<programmiSerata giorno="2002-02-25">
  <rete nome="Rai 1">
    <programma>
      <titolo>TG 1</titolo>
      <tipologia>Notiziario</tipologia>
      <inizio>20:30</inizio>
    </programma>
    <programma>
      <titolo>Affari Tuoi</titolo>
      <tipologia>Gioco</tipologia>
      <descrizione>Gioco a premi con i pacchi</descrizione>
      <dalle>20:30</dalle>
      <alle>21:10</alle>
    </programma> ...
```

Tecnologie Web - 55



### DTD - 1

```
<!DOCTYPE programmiSerata [
  <!ELEMENT programmiSerata ( rete ) >
  <!ELEMENT rete ( programma+ ) >
  <!ATTLIST rete nome CDATA #REQUIRED >
  <!ELEMENT programma ( titolo,tipologia, puntata?, descrizione?,
    ((dalle,alle) | inizio) ) >

  <!ELEMENT titolo ( #PCDATA ) >
  <!ELEMENT tipologia ( #PCDATA ) >
  <!ELEMENT puntata ( #PCDATA ) >
  <!ELEMENT descrizione ( #PCDATA ) >
  <!ELEMENT dalle ( #PCDATA ) >
  <!ELEMENT alle ( #PCDATA ) >
  <!ELEMENT inizio ( #PCDATA ) >
]>
```

Tecnologie Web - 56



### DTD - 2

```
<!ATTLIST programmiSerata giorno CDATA #REQUIRED
  xmlns CDATA #FIXED "http://www.programmimiTV.it"
  xmlns:xs CDATA #FIXED
    "http://www.w3.org/2001/XMLSchema-instance"
  xs:schemaLocation CDATA #FIXED
    "http://www.programmimiTV.it tv.xsd" >
```

Tecnologie Web - 57



### XMLSchema

```
<xs:element name="programmiSerata">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="rete" maxOccurs="unbounded"
        type="Trete"/>
    </xs:sequence>
    <xs:attribute name="giorno" type="xs:date"/>
  </xs:complexType>
</xs:element>
```

Tecnologie Web - 58



### XMLSchema

```
<xs:complexType name="Trete">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="programma"
      type="Tprogrammi"/>
  </xs:sequence>
  <xs:attribute name="nome" type="xs:string" />
</xs:complexType>
<xs:complexType name="Tprogrammi">
```

Tecnologie Web - 59



### 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"
      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>
```

Tecnologie Web - 60



## XMLSchema

```
<xs:simpleType name="Tipologia">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Notiziario"/>
    <xs:enumeration value="Gioco"/>
    <xs:enumeration value="Spettacolo"/>
    <xs:enumeration value="Film"/>
    <xs:enumeration value="Approfondimento"/>
    <xs:enumeration value="Previsioni del tempo"/>
  </xs:restriction>
</xs:simpleType>
```

Tecnologie Web - 61

## Esercizio 5

- 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)

Tecnologie Web - 62

## Esercizio 5 - Documento XML

```
<?xml version="1.0" encoding="UTF-8"?>
<shiporder orderid="889923" xmlns ...>
  <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>
    <price>9.90</price>
  </item>
</shiporder>
```

Tecnologie Web - 63

## 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

Tecnologie Web - 64

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

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="shiporder">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="orderperson" type="xs:string"/>
        <xs:element name="shipto">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="address" type="xs:string"/>
              <xs:element name="city" type="xs:string"/>
              <xs:element name="country" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

Tecnologie Web - 65

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

```
<xs:element name="item" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="note" type="xs:string"
        minOccurs="0"/>
      <xs:element name="quantity" type="xs:positiveInteger"/>
      <xs:element name="price" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:attribute name="orderid" type="xs:string" use="required"/>
</xs:schema>
```

Tecnologie Web - 66

### 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 - Fette di Salame - 2

```
<xs:element name="item">
  <xs:complexType> <xs:sequence>
    <xs:element ref="title"/> <xs:element ref="note" minOccurs="0"/>
    <xs:element ref="quantity"/> <xs:element ref="price"/>
  </xs:sequence> </xs:complexType>
</xs:element>
<xs:element name="name" type="xs:string"/>
<xs:element name="address" type="xs:string"/>
<xs:element name="city" type="xs:string"/>
<xs:element name="country" type="xs:string"/>
<xs:element name="title" type="xs:string"/>
<xs:element name="note" type="xs:string"/>
<xs:element name="quantity" type="xs:positiveInteger"/>
<xs:element name="price" type="xs:decimal"/>
<xs:attribute name="orderid" type="xs:string" />
</xs:schema>
```

Tecnologie Web - 68



### Esercizio 5 - Schema XML - Giardino dell'Eden - 1

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="shiporder" type="Torder"/>
  <xs:element name="orderperson" type="xs:string"/>
  <xs:element name="shipto" type="Tshipto"/>
  <xs:element name="item" type="Titem"/>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="address" type="xs:string"/>
  <xs:element name="city" type="xs:string"/>
  <xs:element name="country" type="xs:string"/>
  <xs:element name="title" type="xs:string"/>
  <xs:element name="note" type="xs:string"/>
  <xs:element name="quantity" type="xs:positiveInteger"/>
  <xs:element name="price" type="xs:decimal"/>
  <xs:attribute name="orderid" type="xs:string"/>
</xs:schema>
```

Tecnologie Web - 69



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

```
<xs:complexType name="Torder">
  <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: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>
```

Tecnologie Web - 70



### Esercizio 5 - Schema XML - Giardino dell'Eden - 3

```
<xs:complexType name="Titem"> <xs:sequence>
  <xs:element ref="title"/>
  <xs:element ref="note" minOccurs="0"/>
  <xs:element ref="quantity"/>
  <xs:element ref="price"/>
</xs:sequence> </xs:complexType>
</xs:schema>
```

Tecnologie Web - 71



### Esercizio 5 - Schema XML - Tende alla Veneziana 1

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="shiporder" type="Torder"/>
  <xs:complexType name="Torder">
    <xs:sequence>
      <xs:element name="orderperson" type="xs:string"/>
      <xs:element name="shipto" type="Tshipto"/>
      <xs:element name="item" type="Titem"
        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="orderid" type="xs:string" use="required"/>
  </xs:complexType>
```

Tecnologie Web - 72



### Esercizio 5 - Schema XML - Tende alla Veneziana 2

```
<xs:complexType name="Tshipto"> <xs:sequence>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="address" type="xs:string"/>
  <xs:element name="city" type="xs:string"/>
  <xs:element name="country" type="xs:string"/>
</xs:sequence> </xs:complexType>
<xs:complexType name="Titem"> <xs:sequence>
  <xs:element name="title" type="xs:string"/>
  <xs:element name="note" type="xs:string" minOccurs="0"/>
  <xs:element name="quantity" type="xs:positiveInteger"/>
  <xs:element name="price" type="xs:decimal"/>
</xs:sequence> </xs:complexType>
</xs:schema>
```

Tecnologie Web - 73



### 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>
```

Tecnologie Web - 74



### 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>
    <note>Special Edition</note>
    <quantity>1</quantity> <price>10.90</price>
  </item>
  <item> <title>Hide your heart</title> <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

Non  
funziona!

Tecnologie Web - 75



### 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>
    <price>9.90</price>
  </item>
</so:shiporder>
```

Tecnologie Web - 76



### 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 5 - Validazione documento XML

- In questo esempio, un documento validato dallo schema progettato secondo il modello Fette di Salame, viene validato anche dallo schema progettato secondo il modello Giardino dell'Eden
- In questo esempio, un documento validato dallo schema progettato secondo il modello Bambole Russe, viene validato anche dallo schema progettato secondo il modello Tende alla Veneziana

Tecnologie Web - 78



### Esercizio 5 - riepilogo

- I modelli Bambole Russe e Tende alla Veneziana permettono di scrivere meno codice
- Riutilizzo
  - Il modello Bambole russe non permette il riutilizzo
  - I modelli Fette di Salame e Giardino dell'Eden permettono di riutilizzare ogni elemento (o attributo) definito
  - Il modello Tende alla Veneziana permette il riutilizzo dei tipi e non degli elementi

Tecnologie Web - 79



### Esercizio 6

- Creare un documento istanza XML che descriva un curriculum con:
    - Dati personali (Nome e cognome, data di nascita e residenza)
    - Esperienze formative
    - Esperienze professionali
- e uno schema che lo validi

Tecnologie Web - 80



### Esercizio 6 - documento XML

```
<cv:curriculumVitae ...>
  <nomeCognome>Mario Rossi</nomeCognome>
  <dataNascita>01/01/1960</dataNascita>
  <residenza>Via Verdi, 9 Padova (PD)</residenza>
  <espFormative tipo="diploma">
    <data>01/01/1980</data>
    <descr>Liceo bla bla bla...</descr>
  </espFormative>
  <espFormative tipo="laurea">
    <data>01/01/1990</data>
    <descr>Laurea in bla bla bla...</descr>
  </espFormative>
  <espProf>
    <periodo>01/01/1991 - 01/01/1998</periodo>
    <descr>bla bla bla...</descr>
  </espProf>
</cv:curriculumVitae>
```

Tecnologie Web - 81



### Esercizio 6 - schema 1

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.dominio.com"
  xmlns="http://www.dominio.com">
  <xs:element name="curriculumVitae" type="Tcv"/>
  <xs:complexType name="Tcv"><xs:sequence>
    <xs:choice>
      <xs:element name="nomeCognome" type="xs:string" />
      <xs:sequence>
        <xs:element name="nome" type="xs:string" />
        <xs:element name="cognome" type="xs:string" />
      </xs:sequence>
    </xs:choice>
    <xs:element name="dataNascita" type="Tdata"/>
    <xs:element name="residenza" type="xs:string"/>
    <xs:element name="espFormative" type="TespForm"
      maxOccurs="unbounded"/>
    <xs:element name="espProf" type="TespProf" maxOccurs="unbounded"/>
  </xs:sequence></xs:complexType>
```

Tecnologie Web - 82



### Esercizio 6 - schema 2

```
<xs:simpleType name="Tdata">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{2}/[0-9]{2}/[0-9]{4}" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="TespForm">
  <xs:sequence>
    <xs:element name="data" type="Tdata"/>
    <xs:element name="descr" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="tipo" type="xs:string"/>
</xs:complexType>
<xs:complexType name="TespProf">
  <xs:sequence>
    <xs:element name="periodo" type="xs:string"/>
    <xs:element name="descr" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Tecnologie Web - 83



Tecnologie Web - 84



### 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** è 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.

Tecnologie Web - 85



### Esercizio 4 - 1

- Dato il seguente schema produrre un documento valido

```
<xsd:simpleType name="carattere">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[A-Z]" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="cifra">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="car" type="carattere" />
<xsd:element name="num" type="cifra" />
```

Tecnologie Web - 86



### 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" maxOccurs="2" />
    <element ref="car" />
    <element ref="num" minOccurs="2" maxOccurs="2" />
    <element ref="car" />
    <element ref="num" minOccurs="3" maxOccurs="3" />
    <element ref="car" />
  </xsd:sequence>
</xsd:complexType>
```

Tecnologie Web - 87



### Esercizio 4 - soluzione

```
<esercizio>
  <car>X</car><car>X</car><car>X</car>
  <car>Y</car><car>Y</car><car>Y</car>
  <num>1</num> <num>1</num>
  <car>Z</car>
  <num>2</num> <num>2</num>
  <car>H</car>
  <num>3</num> <num>3</num> <num>3</num>
  <car>K</car>
</esercizio>

XXXYYY11Z22H333K
```

Tecnologie Web - 88

