

## Esercizi su XSL e XSLT

**Esercizio 1** Scrivere un foglio di stile XSL che visualizza come testo HTML i nomi degli elementi XML del documento di input.

### Soluzione

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">
  <xsl:output method="html"/>

  <xsl:template match="/*">
    <html>
      <body>
        <ul>
          <li><xsl:value-of select="name()"/></li>
          <xsl:apply-templates select="*"/>
        </ul>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="*">
    <li><xsl:value-of select="name()"/></li>
    <xsl:apply-templates select="*"/>
  </xsl:template>

</xsl:stylesheet>
```

**Esercizio 2** Scrivere un foglio di stile XSL che fa diventare tutti gli elementi XML del documento di input figli della radice.

### Soluzione

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">
  <xsl:output method="xml"/>
```

```

<xsl:template match="/*">
  <xsl:element name="{name()}">
    <xsl:apply-templates/>
  </xsl:element>
</xsl:template>

<xsl:template match="*">
  <xsl:element name="{name()}">
    </xsl:element>
    <xsl:apply-templates/>
  </xsl:template>

</xsl:stylesheet>

```

**Esercizio 3** Scrivere un foglio di stile XSL che fa diventare tutti gli elementi XML del documento di input figli della radice, mantenendo i nodi testo nella giusta posizione.

### Soluzione

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="xml"/>

  <xsl:template match="/*">
    <xsl:element name="{name()}">
      <xsl:for-each select="text()">
        <xsl:copy/>
      </xsl:for-each>
      <xsl:apply-templates select="*" />
    </xsl:element>
  </xsl:template>

  <xsl:template match="*">
    <xsl:element name="{name()}">
      <xsl:for-each select="text()">
        <xsl:copy/>
      </xsl:for-each>
    </xsl:element>
    <xsl:apply-templates select="*" />
  </xsl:template>

</xsl:stylesheet>

```

**Esercizio 4** Scrivere un foglio di stile XSL che genera un documento HTML contenente la lista ordinata di tutte le parti di testo contenute nel documento XML di input, utilizzando delle sottoliste puntate per elencare tutte le parti di testo relative a uno stesso elemento.

### Soluzione

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">
  <xsl:output method="html"/>

  <xsl:template match="/*">
    <html>
    <body>
    <ol>
      <li>
        contenuto testuale di <xsl:value-of select="name()"/>:
        <ul>
          <xsl:apply-templates select="text()"/>
        </ul>
      </li>
      <xsl:apply-templates select="*"/>
    </ol>
    </body>
    </html>
  </xsl:template>

  <xsl:template match="*">
    <li>
      contenuto testuale di <xsl:value-of select="name()"/>:
      <ul>
        <xsl:apply-templates select="text()"/>
      </ul>
    </li>
    <xsl:apply-templates select="*"/>
  </xsl:template>

  <xsl:template match="text()">
    <li>
      <xsl:copy/>
    </li>
  </xsl:template>

</xsl:stylesheet>
```

**Esercizio 5** Scrivere un foglio di stile XSL che, dato un documento XML, restituisce il documento tale che: 1) l'elemento root e' uguale all'elemento root del documento di input; 2)

ogni elemento <x> diverso dall'elemento radice viene trasformato in un elemento <nuovo>, con valore dell'attributo elem uguale a <x> (cioè <nuovo elem="x">).

Ad esempio, se il documento XML di input è il seguente:

```
<root>
  <b>
    <a>testo1</a>
    <c>
      <a/>
      testo2
      <d>
        <a>testo3</a>
      </d>
    </c>
    testo4
    <a/>
  </b>
</root>
```

il foglio di stile applicato al documento deve restituire il documento seguente:

```
<root>
  <nuovo elem="b">
    <nuovo elem="a">testo1</nuovo>
    <nuovo elem="c">
      <nuovo/>
      testo2
      <nuovo elem="d">
        <nuovo elem="a">testo3</nuovo>
      </nuovo>
    </nuovo>
    testo4
    <nuovo elem="a"/>
  </nuovo>
</root>
```

## Soluzione

```
<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="xml"/>

  <xsl:template match="/*">
    <xsl:element name="{name()}">
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

```
    </xsl:element>
  </xsl:template>

  <xsl:template match="*">
    <xsl:element name="nuovo">
      <xsl:attribute name="elem">
        <xsl:value-of select="name()"/>
      </xsl:attribute>
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>

</xsl:stylesheet>
```