

# Codifica TEI - Visualizzazione ed Elaborazione XML

Angelo Mario Del Grosso

..

`angelo.delgrosso@ilc.cnr.it`

*CNR-ILC-LicoLab*

`http://licolab.ilc.cnr.it/`

Istituto di Linguistica Computazionale “A. Zampolli”,  
26th November 2018

# Sommario della Lezione

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

**1** Introduzione

**2** XSLT: Documenti Multipli e Versione 2.0

**3** XML Transformations: Un esempio con XML-TEI

**4** Panoramica sul Document Object Model (DOM)

**5** XML DOM Programming API: Un esempio TEI

# Progress status

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## 1 Introduzione

## 2 XSLT: Documenti Multipli e Versione 2.0

## 3 XML Transformations: Un esempio con XML-TEI

## 4 Panoramica sul Document Object Model (DOM)

## 5 XML DOM Programming API: Un esempio TEI

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Perché visualizzare e elaborare il testo

- Controllare la codifica e correggere i refusi
- Assicurarsi che tutto sia stato trascritto correttamente
- Mostrare il testo a persone che non conoscono XML-TEI
- Disporre di una versione del lavoro fruibile
- Manipolare e analizzare le informazioni codificate

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## I fogli di stile (style sheet)

- Descrive il modo in cui un documento elettronico deve essere visualizzato
- Il mezzo di visualizzazione può variare: lo schermo di un computer, la stampa, i sintetizzatori vocali, ecc.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello del Documento

- Descrive il modo in cui un documento elettronico deve essere rappresentato e navigato da agenti software
- Indipendenza dal linguaggio e dalla piattaforma

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Scopo dei fogli di stile

- Separazione forma-contenuto: *la visualizzazione del documento è un processo indipendente (e successivo)*
- Gestione della resa grafica per molti documenti contemporaneamente: *massima uniformità dello stile*
- Gestione di mezzi diversi dal monitor: smartphone, sintetizzatore vocale, stampante braille, ecc.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Scopo di un modello per i documenti

- Astrazione: *la navigazione e la elaborazione sono spaccificate da uno standard e non cambia nelle sue varie implementazioni*
- Uniformità: *le interfacce sono comuni a tutti i sistemi che implementano il modello*
- Efficacia ed Efficienza: il modello identifica oggetti, metodi e proprietà utili alla gestione e al controllo programmatico dei documenti.



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Metodi e Tecnologie

**Quelli più noti e utilizzati sono standard internazionali definiti dal consorzio W3**

(<http://www.w3.org/Style/CSS>).

- XSL: eXtensible Stylesheet Language
- DOM: Document Object Model

# Progress status

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

1 Introduzione

2 XSLT: Documenti Multipli e Versione 2.0

3 XML Transformations: Un esempio con XML-TEI

4 Panoramica sul Document Object Model (DOM)

5 XML DOM Programming API: Un esempio TEI

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: Modularità

- XSLT consente di importare ed includere fogli XSLT dentro altri documenti XSLT
- mediante gli elementi `<xsl:import>` ed `<xsl:include>`

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation

- **inclusione:** le regole definite nel documento incluso hanno la stessa priorità delle regole definite nel documento XSLT principale.
- **importazione:** le regole definite nel documento principale hanno una priorità maggiore.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Dichiarazione XSD dell'elemento xsl:include

```
<xs:element name="include" substitutionGroup="xsl:declaration">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xsl:element-only-versioned-element-type">
        <xs:attribute name="href" type="xs:anyURI" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: `xsl:include`

Dato un foglio esterno `esterno.xml` per includerlo nel foglio principale si usa l'elemento:

```
<xsl:include href="esterno.xml"/>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Dichiarazione XSD dell'elemento `xsl:import`

```
<xsl:element name="import">
  <xsl:complexType>
    <xsl:complexContent>
      <xsl:extension base="xsl:element-only-versioned-element-type">
        <xsl:attribute name="href" type="xs:anyURI" use="required"/>
      </xsl:extension>
    </xsl:complexContent>
  </xsl:complexType>
</xsl:element>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: `xsl:import`

```
<xsl:import href="esterno.xsl"/>
<xsl:template match="/" >
  <xsl:apply-imports/>
</xsl:template>
```



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: documenti multipli

- Accedere ai nodi di un documento XML esterno a quello che si sta elaborando
- Utilizzando la funzionalità `document()`

## XML Style Sheet Transformation: documenti multipli

```
<xsl:value-of  
  select="document('altro.xml')/TEI/text/@type"  
>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: XSLT 2.0

La *recommendation* W3C per XSLT 2.0 è stata pubblicata nel 2007

## XML Style Sheet Transformation: documenti multipli

```
<xsl:stylesheet  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="2.0" >
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: XSLT 2.0

XSLT 2.0 permette di creare un documento di output in vari formati (xml, html, xhtml, text).

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XS<sub>LT</sub>:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: documenti multipli

```
<xsl:template match="/" >  
  <xsl:result-document method="html"  
href="output.html" >  
    <xsl:apply-templates/>  
  </xsl:result-document>  
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: XSLT 2.0

XSLT 2.0 permette di creare anche documenti multipli.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: XSLT 2.0 esempio

```
<template match="//div/[@type='book']" >
  <xsl:for-each select="div/[@type='chapter']" >
    <xsl:result-document
      method="html"
      href="{@n}.html" >
      <xsl:apply-templates/>
    </xsl:result-document>
  </xsl:for-each>
</template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: XSLT 2.0

- elemento `<xsl:for-each-group>`: permette di selezionare un set di items
- processare un gruppo per volta

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XML Style Sheet Transformation: XSLT 2.0

### *Possibili attributi*

- group-by
- group-adjacent
- group-starting-with
- group-ending-with



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Esempio elemento for-each-group*

```
<xsl:for-each-group select="impiegato" groupby="@dipartimento">
  <dipartimento>
    <nome>
      <xsl:text>Dipartimento</xsl:text>
      <xsl:value-of select="current-grouping-key()" />
    </nome>
    <xsl:for-each select="current-group()">
      <persona><xsl:value-of select="node()" /></persona>
    </xsl:for-each>
  </dipartimento>
</xsl:for-each-group>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Espressioni condizionali if()*

**XSLT 1.0**

```
<xsl:variable name="prezzo">
  <xsl:choose>
    <xsl:when test="discount">
      <xsl:value-of select="price - discount"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="price"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

**XSLT 2.0**

```
<xsl:variable name="prezzo"
  select="if(discount) then (price - discount) else(price)"/>
```

# Progress status

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

1 Introduzione

2 XSLT: Documenti Multipli e Versione 2.0

**3 XML Transformations: Un esempio con XML-TEI**

4 Panoramica sul Document Object Model (DOM)

5 XML DOM Programming API: Un esempio TEI

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

*Tutti i nomi di elementi TEI devono essere preceduti dal prefisso **tei**:*

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

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:tei="http://www.tei-c.org/ns/1.0"
  xmlns="http://www.w3.org/1999/xhtml">
  .
  .
  .
  <xsl:output
    method="html"
    encoding="UTF-8"
    indent="yes"
    omit-xml-declaration="yes" />
  .
  .
  .
</xsl:stylesheet>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

*Creare lo “scheletro” del documento finale, nel nostro caso un HTML*

```
<xsl:template match="/tei:TEI">
  <html>
    <head>
      <title>*-* Titolo del documento *-*</title>
      <link href="stile.css" rel="stylesheet" type="text/css"/>
    </head>
    <body>
      <xsl:apply-templates />
    </body>
  </html>
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

*Definire una regola per l'intestazione TEI (<teiHeader>)*

```
<xsl:template match="tei:teiHeader">  
  <h1>Informazioni sul documento</h1>  
  <xsl:apply-templates />  
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Componenti dell'elemento **tei:text***

```
<xsl:template match="tei:text/tei:front">
  <h1>Frontespizio e prefazione</h1> <br/>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="tei:text/tei:body">
  <h1>Testo</h1> <br/>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="tei:text/tei:back">
  <h1>Appendici</h1> <br/>
  <xsl:apply-templates/>
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

## *Divisioni del testo*

```
<xsl:template match="tei:div" mode="front">
  . . . . . <br /><hr width="75%" /><br />
  . . . . . <xsl:apply-templates />
</xsl:template>

<xsl:template match="tei:div">
  . . . . . <p><xsl:apply-templates /></p>
</xsl:template>

<xsl:template match="tei:byline">
  . . . . . <p><xsl:apply-templates /></p>
</xsl:template>
```



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

*Per la poesia*

```
<xsl:template match="tei:lg">
  <p>
    <xsl:if test="@">
      <b><xsl:value-of select="@n" /></b>
    </xsl:if>
  </p>
</xsl:template>

<xsl:template match="tei:l">
  <xsl:apply-templates /><br/>
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Elementi Phrase-Level*

```
<xsl:template match="//tei:div/tei:head">
  <h3 class="center"><xsl:apply-templates /></h3>
</xsl:template>

<xsl:template match="tei:term">
  <b><i><xsl:apply-templates /></i></b>
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Interventi Editoriali*

```
<xsl:template match="tei:sic">
|   . . . <xsl:apply-templates />
</xsl:template>
.

<xsl:template match="tei:corr">
|   . . . [<xsl:apply-templates />]
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Interventi Editoriali*

```
<xsl:template match="tei:orig">
  . . . [<xsl:apply-templates />]
</xsl:template>
.
<xsl:template match="tei:reg">
  . . . [<xsl:apply-templates />]
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Altri elementi*

```
<xsl:template match="tei:emph">
  . . . . . <b><xsl:apply-templates/></b>
</xsl:template>

<xsl:template match="tei:distinct">
  . . . . . <font FACE="Garamond, Times, serif">
  . . . . . | . . . . . <xsl:apply-templates/>
  . . . . . </font>
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

*Elemento <hi> distinto in base all'attributo @rend*

```
<xsl:template match="tei:hi">
  <xsl:choose>
    <xsl:when test="@rend='italic'">
      <i><xsl:apply-templates/></i>
    </xsl:when>
    <xsl:when test="@rend='bold'">
      <b><xsl:apply-templates/></b>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Elementi vuoti e milestone*

```
<xsl:template match="tei:lb">
  <br/><xsl:if test="./@n">
    <span class="line">
      <xsl:value-of select="current()/@n" />
    </span>
  </xsl:if>
</xsl:template>
```

# Progress status

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

1 Introduzione

2 XSLT: Documenti Multipli e Versione 2.0

3 XML Transformations: Un esempio con XML-TEI

4 **Panoramica sul Document Object Model (DOM)**

5 XML DOM Programming API: Un esempio TEI



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Documenti Object Model (DOM)

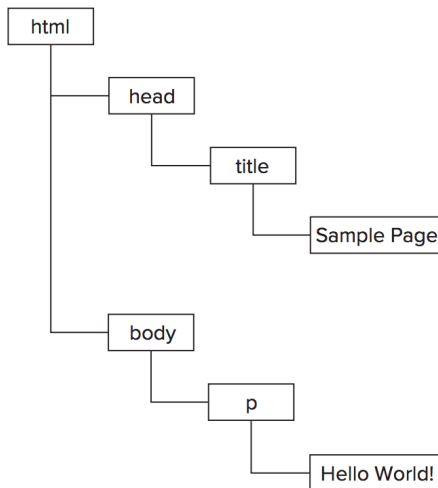
Il **Document Object Model** (*DOM*) è un modello *language-independent* che mette a disposizione delle **Application Programming Interface** (*API*) per accedere e manipolare documenti XML (e HTML).

## Documenti Object Model (DOM)

Il Modello DOM rappresenta l'intero documento XML come una gerarchia di nodi (*albero*).

# Visualizzare ed Elaborare documenti XML

## *Esempio albero DOM*



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Documenti Object Model (DOM)

Ciascuna parte di un documento XML-TEI è quindi un tipo specifico di nodo DOM costituito da diversi tipi di dati.

## Documenti Object Model (DOM)

Grazie alla rappresentazione gerarchica messa a disposizione dal DOM è possibile avere un livello di controllo a granularità molto fine per manipolare quasi tutti gli aspetti del documento XML originario.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Documenti Object Model (DOM): espressività

I nodi di un albero DOM possono essere aggiunti, modificati, cancellati o rimpiazzati facilmente grazie alle API definite dal modello.

## Documenti Object Model (DOM)

Il modello DOM è una specifica e una raccomandazione del consorzio di standardizzazione Web W3C.  
<https://www.w3.org/DOM/DOMTR>

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Documenti Object Model (DOM): Browser-Independent

Una importante caratteristica del Modello e delle API DOM è la sua indipendenza dalla tecnologia di implementazione.

## Tecnologia indipendente dal software

Un importante obiettivo dello standard DOM è quello di essere cross-platform.

In realtà però non tutti i browser implementano con le stesse modalità le specifiche, così come non tutti i browser sono allineati all'ultima versione del modello (copertura non al 100%).

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## DOM Levels

Esistono diverse versioni (livelli) dello standard DOM nonché diverse moduli all'interno di ciascun livello.

## DOM Levels

Allo stato attuale, il Modello DOM conta 5 diverse versioni o approcci (livelli), a loro volta suddivisi in sezioni o aree.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## le API con il linguaggio di programmazione javascript

E' possibile sfruttare, anche attraverso javascript nativo, la rappresentazione ad albero di un documento XML e le funzionalità che DOM mette a disposizione per accedere, manipolare ed estrarre informazioni da un documento XML.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## DOM Level 0

**Il livello 0** del DOM indica la modalità 'legacy' di operare sui documenti, sarebbe cioè quella che non tiene di conto delle specifiche W3C.

## DOM Level 0

Adottando DOM Level 0 si suppongono anche scelte implementative non cross-platform e non browser-independent.



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## DOM Level 1

**DOM Level 1** è la prima versione dello standard  
(*raccomandazione W3C nel 1998*).

La specifica definisce **oggetti, proprietà e funzionalità di base** per navigare e manipolare la struttura di un documento XML (e HTML).

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## DOM Level 1

Il DOM Level 1 è diviso in due sezioni:

- 1 **Generica** per la gestione di documenti XML e HTML
- 2 **Specifica** per documenti HTML: questa sezione estende la prima (Sezione Generale).

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## DOM Level 2

**DOM Level 2** è una estensione del livello 1 e aggiunge a quest'ultimo numerose funzionalità e moduli specifici, come per esempio la gestione degli eventi e dei fogli di stile CSS.

## DOM Level 2

Tra i vari moduli che *DOM Level 2* aggiunge, di particolare interesse è il modulo **DOM Traversal and Range**.

**Lo standard DOM Level 3 ha raggiunto lo status di raccomandazione W3C nel 2004.**

## DOM Level 3

- Migliora ed estende il modulo di gestione degli eventi definito nel DOM Level 2
- Aggiunge funzionalità per *supportare tutta la specifica della versione 1.0* di XML (es.: **XML namespace**, **XPath**, **DOM Validaton**)
- Caricamento e salvataggio dei documenti con serializzazione XML (**DOM Load and Save**).

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## DOM Level 4

**DOM Level 4** è la prossima versione del Modello DOM sulla quale sta lavorando il W3C. Ancora non ha raggiunto lo status di raccomandazione e non è ancora supportata dai browser.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Supporto dei browser alle specifiche DOM

Da qualche anno il supporto alle specifiche e alle API DOM è **diventato una priorità**, oltre che una necessità, per i più grossi gruppi che sviluppano browser.

Tuttavia non tutte le principali compagnie implementano allo stesso modo lo standard.

## Il Modello DOM in sintesi

- Un documento XML può essere rappresentato come una gerarchia di nodi attraverso il modello DOM.
- Ciascun tipo di nodo ha caratteristiche, dati e funzionalità differenti.
- Ciascun tipo di nodo può avere differenti tipi di relazioni con gli altri tipi di nodi.
- Le relazioni tra i nodi DOM creano una gerarchia, la quale permette di rappresentare un documento XML come un albero.

## Il Modello DOM: principali caratteristiche

- Il Modello DOM ha un nodo **document** che rappresenta tutto il documento attraverso la cosiddetta **radice** del documento.
- Il Modello DOM ha un elemento detto *document element* che è l'elemento più esterno del documento, il quale è avo di tutti gli altri elementi e che quindi li racchiude tutti. (esiste solo un *document element* per documento)
- Ciascun pezzo del documento XML può essere rappresentato da un nodo DOM in un albero DOM.



## Il Modello DOM: principali caratteristiche

- A differenza di XSLT, DOM presenta 12 node types, ciascuno dei quali eredita da un nodo di base (in js tutto il data model parte dall'interfaccia *Node Type*).
- Ciascun nodo ha una proprietà che indica il tipo del nodo stesso (in js `nodeType`)

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *I tipi di nodo nel modello DOM*

```
Node.ELEMENT_NODE; // 1
Node.ATTRIBUTE_NODE; // 2
Node.TEXT_NODE; // 3
Node.CDATA_SECTION_NODE; // 4
Node.ENTITY_REFERENCE_NODE; // 5
Node.ENTITY_NODE; // 6
Node.PROCESSING_INSTRUCTION_NODE; // 7
Node.COMMENT_NODE; // 8
Node.DOCUMENT_NODE; // 9
Node.DOCUMENT_TYPE_NODE; // 10
Node.DOCUMENT_FRAGMENT_NODE; // 11
Node.NOTATION_NODE; // 12
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: principali caratteristiche

- Tutti i nodi in un documento DOM instaurano relazioni gerarchiche tra di loro.
- Ciascun nodo ha una proprietà *childNodes* contenente una lista di nodi, detta *NodeList*.

## Il Modello DOM: principali caratteristiche

Un oggetto *NodeList* è simile ad un array usato per memorizzare una lista ordinata di nodi accessibili per posizione

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

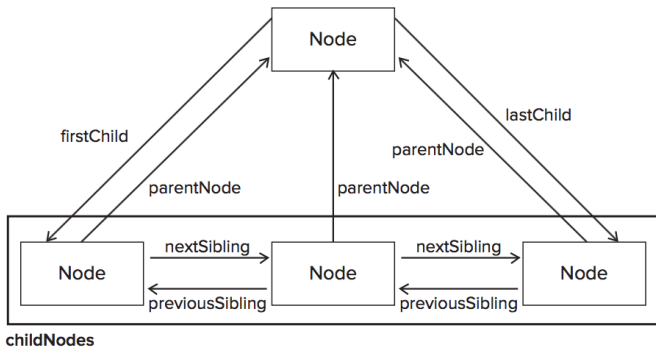
XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: esempio NodeList

```
firstChild = someNode.childNodes[0];  
secondChild = someNode.childNodes.item(1);  
count = someNode.childNodes.length;
```

# Visualizzare ed Elaborare documenti XML

## Le relazioni di un nodo



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: proprietà `ownerDocument`

Tutti i nodi condividono la proprietà `ownerDocument` che è un collegamento diretto al nodo che rappresenta l'intero documento.

## Il Modello DOM: proprietà `ownerDocument`

La proprietà `ownerDocument` fornisce un modo veloce per accedere al *nodo documento* senza dover attraversare tutta la gerarchia di nodi per arrivare alla radice dell'albero.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: metodi per la manipolazione

Una serie di metodi sono disponibili per la manipolazione dei nodi dell'albero DOM.

## Il Modello DOM: metodi per la manipolazione

Quattro di questi metodi (sei in totale) lavorano sui figli di uno specifico nodo.

Poiché non tutti i tipi di nodi hanno nodi figli, questi ultimi metodi possono lanciare degli errori se invocati.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## *Il Modello DOM: metodi per la manipolazione*

```
node.appendChild(newChild);  
node.insertBefore(newChild, refChild);  
node.replaceChild(newChild, oldChild);  
node.removeChild(oldChild);  
  
node.cloneNode(deep);  
.  
node.normalize();
```



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: Document Type

- Rappresenta un intero documento ed è la radice della gerarchia.
- L'oggetto *document* è una istanza di tipo **Document**, attraverso di esso è possibile interrogare e ottenere altri nodi in molti modi diversi.

## Il Modello DOM: Document Type esempio

```
xmlRootElement = document.documentElement;  
divElement = document.getElementById('divID');  
myDivList = document.getElementsByTagName('div');
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: Element Type

- Rappresenta elementi XML (ma anche HTML) in un documento
- Usato per accedere e manipolarne il contenuto e/o il valore dei propri attributi

## Il Modello DOM: Element Type esempio

```
myDiv.tagName == myDiv.nodeName  
if (element.tagName.toLowerCase() == 'div')//  
istruzioni varie
```

## Il Modello DOM: Attribute Type

- Tecnicamente anche gli attributi sono nodi.
- Possono essere acceduti attraverso la proprietà *attributes* di un oggetto *element*.
- I nodi figlio di un attributo possono essere nodi di tipo *Text* oppure nodi di tipo *EntityReference*.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: Attribute Type esempio

```
myDiv.getAttribute('lang');  
attr = document.createAttribute("rend");  
attr.value = "above";  
element.setAttributeNode(attr);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: Text Type

- Un oggetto istanza di tipo *Text* contiene testo puro che viene interpretato come *literal*.
- Può contenere caratteri XML con escape, ma non può contenere codice XML.
- Gli elementi che contengono testo hanno almeno un nodo di tipo testo.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Il Modello DOM: Text Type esempio

```
var textNode = myDiv.firstChild;  
var text = textNode.nodeValue;  
var newTextNode = document.createTextNode(' testo  
aggiunto');  
myDiv.appendChild(newTextNode);  
myDiv.normalize();  
element.firstChild.splitText(5);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM per la manipolazione di documenti XML

I moduli Core di DOM Level 2 e 3 estendono il modello e le API DOM in modo da coprire tutte le specifiche XML.

## Verificare con js se il browser implementa i moduli XML del modello DOM

```
document.implementation.hasFeature("\Core", \2.0");  
document.implementation.hasFeature("\Core", \3.0");  
document.implementation.hasFeature("\XML", \2.0");
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM verificare lo stato dell'implementazione

L'oggetto *document* ha una proprietà, *implementation*, che presenta informazioni e funzionalità direttamente collegate all'implementazione DOM dei vari browser.

## Modello DOM verificare lo stato dell'implementazione

Tramite questa proprietà è possibile testare se sono implementate o meno le varie sezioni dello standard. Non è detto però che a tale verifica corrisponda effettivamente la corretta implementazione del modulo.



## Modello DOM per XML: Level 2 e Level 3

- Il modulo DOM Level 2 Core introduce nuove funzionalità collegate alla gestione dei namespaces XML.
- DOM Level 2 permette la creazione di nuove istanze di Document e permette la creazione di oggetti *DocumentType*.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM per XML: Traversals e Range

- I moduli Traversals e Range implementano diverse modalità per elaborare e accedere alla struttura DOM.
- Gli oggetti più importanti sono NodeIterator e TreeWalker per eseguire "visite" **depth-first** dell'albero DOM.

## Modello DOM per XML: Traversals e Range

- Gli oggetti *Ranges* implementano funzionalità per selezionare specifiche porzioni della struttura del DOM per elaborare il documento in qualche modo.
- Per esempio può essere usato per rimuovere porzioni di documento mantenendo sempre una coerenza sintattica e di forma del documento XML.

## Modello DOM per XML

- DOM Level 2 introduce il concetto di creazione dinamica di documenti XML
- DOM Level 3 introduce meccanismi di parsing e di serializzazione XML.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: creazione di documenti

```
xmlDom = document.implementation.createDocument(  
    namespaceUri, root, doctype);  
teiDom = document.implementation.createDocument(  
    'http://www.tei-c.org/ns/1.0', 'tei:TEI',  
    TEIDoctype);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM per XML

- Una volta creato il documento è possibile sfruttare tutte le features che ci mette a disposizione il modello DOM.
- Nuovi nodi possono essere aggiunti al documento e manipolati a piacimento.

## modello DOM per XML: creazione di documenti

```
teiHeaderElem = teiDom.createElement('teiHeader');  
teiDom.documentElement.appendChild(teiHeaderElem);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM per XML

- Più che creare un documento DOM da zero, la funzionalità più utile è quella di *parsare* un documento XML già compilato, oppure la funzionalità inversa di serializzazione.
- Esiste un oggetto DOMParser che può "parsare" stringhe di testo che rappresentano documenti XML **well-formed**

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: creazione di documenti

```
parser = new DOMParser();  
teiDoc =  
parser.parseFromString('<TEI><teiHeader/></TEI>',  
    'text/xml');  
textElement = teiDoc.createElement("text");  
teiDoc.documentElement.appendChild(textElement);
```



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM per XML

- Esiste anche l'oggetto duale di DOMParser. Questo è l'oggetto XMLSerializer.
- Per ottenere una stringa che rappresenti il documento XML bisogna creare una istanza di XMLSerializer e passare al metodo `serializeToString()` il documento DOM.

## modello DOM per XML: serializzazione di documenti

```
serializer = new XMLSerializer();  
xml = serializer.serializeToString(xmlDOM);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Modello DOM per XML

- Non è disponibile un metodo diretto per caricare XML da server o da file in locale.
- Solo Internet Explorer ha sviluppato nativamente una funzionalità del genere sul DOM  
`xmlDOM.load("example.xml")`.
- E' generalmente accettato utilizzare l'oggetto `XMLHttpRequest` per caricare file da percorsi esterni (remoti oppure locali).

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

*Il più delle volte è meglio utilizzare chiamate asincrone e non sincrone per caricare risorse esterne.*

## modello DOM per XML: caricamento di documenti

```
xhr = new XMLHttpRequest();  
xhr.open('get', 'example.xml', false);  
xhr.send(null);  
if ((xhr.status >= 200 && xhr.status < 300) ||  
    xhr.status == 304)  
    text = xhr.responseText;  else ...
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Xpath

Una API per XPath è stata definita solo a partire dalla specifica DOM Level 3, la quale introduce la *DOM Level 3 XPath recommendation*.

## modello DOM per XML: Xpath

```
supportsXPath =  
document.implementation.hasFeature('XPath',  
'3.0');
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XS<sub>LT</sub>:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Xpath

- Oggetto di tipo `XPathEvaluator`: impiegato per valutare (evaluate) le espressioni `XPath`.
- Oggetto di tipo `XPathResult`: impiegato per gestire il tipo di risultato ottenuto da una operazione di selezione `xpath`.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Xpath

```
result = xmldom.evaluate('employee/name',  
xmldom.documentElement, null,  
XPathResult.ORDERED_NODE_ITERATOR_TYPE, null);  
evaluator.evaluate(expression, contextNode,  
resolver, type, result)
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## XPathResult conta 10 tipi diversi di possibili valori

```
• XPathResult.ANY_TYPE;  
• XPathResult.NUMBER_TYPE;  
• XPathResult.STRING_TYPE;  
• XPathResult.BOOLEAN_TYPE;  
• XPathResult.UNORDERED_NODE_ITERATOR_TYPE;  
• XPathResult.ORDERED_NODE_ITERATOR_TYPE;  
• XPathResult.UNORDERED_NODE_SNAPSHOT_TYPE;  
• XPathResult.ORDERED_NODE_SNAPSHOT_TYPE;  
• XPathResult.ANY_UNORDERED_NODE_TYPE;  
• XPathResult.FIRST_ORDERED_NODE_TYPE;
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Xpath Esempio

```
result = xmldom.evaluate(  
    'text/back', xmlTEI.documentElement,  
    null, XPathResult.BOOLEAN_TYPE, null);
```

## modello DOM per XML: Xpath Esempio

```
result = xmldom.evaluate(  
    'count(text/body/div)', xmldom.documentElement,  
    null, XPathResult.NUMBER_TYPE, null);
```



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Namespace support

- Spesso i documenti XML fanno uso di namespace per definire il corretto vocabolario impiegato.
- In questa circostanza `XPathEvaluator` deve essere opportunamente impostato.
- Nei documenti TEI-XML tutti gli elementi sono parte del namespace `http://www.tei-c.org/ns/1.0`

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Namespace support

- Un modo per gestire il namespace è attraverso la creazione di un oggetto `XPathNSResolver`
- Invocando il metodo `createNSResolver()` del documento.
- Sfruttando una funzione che, dato un prefisso, restituisce l'URI del namespace. (`lookupNamespaceURI`)

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: Namespace support example

```
nsresolver =  
xmldom.createNSResolver(xmldom.documentElement);  
result = xmldom.evaluate(  
    "/tei:TEI/tei:text/tei:front/tei:titlePage",  
    xmldom.documentElement, nsresolver,  
    XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: XSLT support

- Esiste un oggetto che può essere usato per elaborare fogli di stile XSLT in javascript: `XSLTProcessor`.
- Il primo passo è caricare due documenti DOM: un XML di base, e l'altro XSLT.

## modello DOM per XML: XSLT support example

```
processor = new XSLTProcessor()  
processor.importStylesheet(xsltdom);  
result = processor.transformToDocument(xmlldom);
```

## modello DOM per XML: XSLT support

- Accanto al metodo `transformToDocument()` esiste anche il metodo `transformToFragment()`.
- La ragione per la quale è utile l'utilizzo di tale metodo è per coprire eventuali casi in cui il risultato della trasformazione vuole essere aggiunto ad un ulteriore altro documento DOM.
- Tra gli argomenti, infatti, deve essere inserito anche il documento destinatario finale per poter effettuare controlli di consistenza.

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: XSLT support example

```
fragment = processor.transformToFragment(xmlDom,  
document);  
div = document.getElementById('divResult');  
div.appendChild(fragment);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: XSLT support

L'oggetto `XSLTProcessor` permette di gestire e indicare **parametri XSLT**.

## modello DOM per XML: XSLT support example

```
processor = new XSLTProcessor();  
processor.importStylesheet(xsltdom);  
processor.setParameter(null, "message", "contenuto  
del messaggio");  
result = processor.transformToDocument(xmlldom);
```

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: XSLT support

Le istanze dell'oggetto `XSLTProcessor` possono essere riusate per più trasformazioni e con diversi fogli di stile (efficienza computazionale).

## modello DOM per XML: XSLT support example

```
processor.importStylesheet(xsltdom);  
//do some transformations  
processor.reset();  
processor.importStylesheet(xsltdom2);
```



# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## modello DOM per XML: concludendo..

la tecnologia XML è ad oggi ben supportata dai vari browser, anche se alcune implementazioni differiscono in termini di copertura dello standard e di aderenza ad esso. Esistono comunque tecnologie per avere un supporto cross-browser per le più comuni funzionalità.

# Progress status

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

1 Introduzione

2 XSLT: Documenti Multipli e Versione 2.0

3 XML Transformations: Un esempio con XML-TEI

4 Panoramica sul Document Object Model (DOM)

5 XML DOM Programming API: Un esempio TEI

# Visualizzare ed Elaborare documenti XML

Codifica TEI -  
Visualizza-  
zione ed  
Elaborazione  
XML

A.M. Del  
Grosso

Introduzione

XSLT:  
Documenti  
Multipli e  
Versione 2.0

XML Trans-  
formations:  
Un esempio  
con XML-TEI

Panoramica  
sul Document  
Object Model  
(DOM)

XML DOM  
Programming  
API: Un  
esempio TEI

## Documenti Object Model (DOM): Esercizio

Caricare il file testTeiNS.xml con javascript e manipolarlo con DOM e Xpath per costruire una visualizzazione appropriata

## Documenti Object Model (DOM)

Scrivere un foglio di stile e applicarlo con javascript al documento DOM caricando il file testTeiNS.xml