



EBook Gratuito

APPENDIMENTO

xslt

Free unaffiliated eBook created from
Stack Overflow contributors.

#xslt

Sommario

Di.....	1
Capitolo 1: Iniziare con xslt.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Installazione o configurazione.....	2
Esempio XSLT semplice.....	3
Capitolo 2: Variabili in xslt.....	7
Examples.....	7
Variabili Xslt.....	7
Titoli di coda.....	8

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [xslt](#)

It is an unofficial and free xslt ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official xslt.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con xslt

Osservazioni

XSLT (XSL Transformations, Extensible Stylesheet Language Transformations) è un linguaggio di programmazione basato su XML e il cui scopo è quello di manipolare e trasformare XML. È un linguaggio di programmazione funzionale e uno standard W3C. Usando XSLT puoi trasformare uno o più documenti XML in un risultato che potrebbe avere una varietà di formati di output (XML, HTML, testo).

Per capire XSLT - è meglio avere una buona conoscenza pratica di:

1. XML
2. XPath

Senza questi due concetti sottostanti, avresti difficoltà ad imparare XSLT. Innanzitutto, perché un documento XSLT è un documento XML. E in secondo luogo, i documenti XSLT fanno uso di XPath per interrogare il documento XML che viene trasformato.

Lettura consigliata: <https://en.wikipedia.org/wiki/XSLT>

Versioni

Versione	specificazione	Stato	Data di rilascio
XSLT 1.0	https://www.w3.org/TR/xslt	Raccomandazione	1998/08/18
XSLT 2.0	https://www.w3.org/TR/xslt20	Raccomandazione	2007-01-23
XSLT 3.0	https://www.w3.org/TR/xslt-30	Raccomandazione del candidato	2015/11/19

Examples

Installazione o configurazione

XSLT è un linguaggio di programmazione per scopi speciali; è ampiamente utilizzato per trasformare documenti XML in un diverso formato XML, in HTML o in formati basati su testo.

Esistono due versioni principali di XSLT in uso: XSLT 1.0 e XSLT 2.0. XSLT 1.0 è più ampiamente implementato ma ha molte restrizioni e limitazioni rispetto a XSLT 2.0; dovrai decidere quale versione usare. Se un processore XSLT 2.0 è disponibile per l'ambiente scelto, questa è quasi sempre la scelta migliore.

(XSLT 1.0 è uscito nel novembre 1999 e molte implementazioni sono apparse entro un anno o due, sia da fornitori tradizionali come Microsoft, IBM, Sun e Oracle, sia da singoli appassionati che lavoravano nel loro tempo libero. XSLT 2.0 è uscito a gennaio 2007, e molti dei processori 1.0 originali non sono mai stati aggiornati, perché i loro sviluppatori avevano perso interesse. Il più usato processore XSLT 2.0 è Saxon, ma non ha il campo per sé: altri prodotti free-standing includono RaptorXML (di Altova), XmlPrime (da CBCL) e Exselt e i processori 2.0 sono anche disponibili incorporati in WebLogic di IBM, nel server di database XML di MarkLogic e in Intel Accelerator di XML.)

La specifica di XSLT 3.0 è tecnicamente completa (proposta di raccomandazione ad aprile 2017), ma per il momento è considerata al massimo livello: considerala solo se hai bisogno urgente di nuove funzionalità, come streaming, pacchetti, supporto JSON o prova / cattura. Esistono tre implementazioni note: Saxon, Exselt e RaptorXML.

Per iniziare con XSLT hai diverse opzioni:

- Utilizza uno strumento XSLT online. Ce ne sono diversi disponibili (cerca "strumento XSLT online"), un IDE online molto affidabile è <http://xslttransform.net/>. Questo è un buon modo per ottenere una sensazione iniziale per la lingua, ma non vorrai continuare su questa strada una volta che stai facendo un vero sviluppo.
- Usa il motore XSLT integrato in ogni browser. Come con gli strumenti online, questi hanno il vantaggio che non è necessario installare nulla; ma i browser supportano solo XSLT 1.0, supportano solo la conversione da XML a HTML e hanno un supporto debug molto debole. Uno sviluppo molto recente - ma ancora sperimentale - è Saxon-JS, che consente l'esecuzione di XSLT 2.0 (e parti di XSLT 3.0) nel browser.
- Installa un processore XSLT (come Saxon o xsltproc). La maggior parte di questi prodotti può essere invocata utilizzando la riga di comando del sistema operativo o utilizzando un'API per particolari linguaggi di programmazione come Java, C, C# o Python.
- Installare un ambiente di sviluppo XML come Altova XML Spy, oXygen da SyncroSoft o Stylus Studio. Sebbene questa sia un'opzione più costosa, offre molto più ricco supporto per lo sviluppo e capacità di debug.

Qualunque opzione tu scelga, dovresti prima decidere se utilizzare XSLT 1.0 o XSLT 2.0.

Esempio XSLT semplice

Ecco un semplice esempio che utilizza XSLT per convertire i dati in un file XML in una tabella in un file HTML. Puoi usarlo per sperimentare con semplici trasformazioni XSLT.

Prerequisito: installare un Java Runtime Environment e aggiungere il percorso di JRE alla variabile PATH. (Su Windows, la maggior parte degli installer aggiungerà Java al tuo percorso per te.) Se funziona, dovresti essere in grado di aprire una finestra della riga di comando ed eseguire il comando `java -version` e ottenere una stampa di informazioni sul tuo JRE.

1. Scarica il processore Saxon-HE XSLT per Java qui: saxon.sourceforge.net e decomprimilo in qualsiasi punto del tuo computer.

2. In un editor di testo, creare un file denominato `pets.xml` con il seguente codice:

```
<pets>
<petType name="Dogs">
  <pet id="123" name="Sparky" vaccineStatus="vaccinated" healthStatus="healthy"/>
  <pet id="234" name="Sadie" vaccineStatus="unvaccinated" healthStatus="sick"/>
  <pet id="345" name="Herman" vaccineStatus="unvaccinated" healthStatus="unknown"/>
</petType>
<petType name="Cats">
  <pet id="456" name="Cleo" vaccineStatus="vaccinated" healthStatus="healthy"/>
  <pet id="567" name="Janet" vaccineStatus="unvaccinated" healthStatus="healthy"/>
  <pet id="678" name="Furry" vaccineStatus="vaccinated" healthStatus="sick"/>
</petType>
</pets>
```

3. In un editor di testo, creare un file denominato `petTransform.xsl` con il seguente codice:

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

<!-- handle the root XML element -->
<xsl:template match="/">
<html><head>
  <title>Pets that are available for adoption</title>
</head>
<body>
  <xsl:apply-templates/>
</body>
</html>
</xsl:template>

<xsl:template match="pets">
<xsl:apply-templates/>
</xsl:template>

<xsl:template match="petType">
<h2><xsl:value-of select="@name"/></h2>
<table id="{@name}">
  <tr>
    <th colname="id">ID</th>
    <th colname="name">Name</th>
    <th colname="vaccinated">Vaccine status</th>
    <th colname="health">Health status</th>
  </tr>
  <tbody>
    <!-- add a row for each pet in this category -->
    <xsl:for-each select="pet">
      <tr>
        <td colname="id"><xsl:value-of select="@id"/></td>
        <td colname="name"><xsl:value-of select="@name"/></td>
        <td colname="vaccinated"><xsl:value-of select="@vaccineStatus"/></td>
        <td colname="health"><xsl:value-of select="@healthStatus"/></td>
      </tr>
    </xsl:for-each>
  </tbody>
</table>
</xsl:template>

<!-- ignore the content of other tags because we processed them elsewhere -->
```

```
<xsl:template match="*">
<!-- do nothing -->
</xsl:template>

</xsl:stylesheet>
```

4. Apri una finestra della riga di comando e vai alla cartella con i file XML e XSLT.

5. Eseguire il comando seguente, dove `path_to_saxon.jar` è il percorso completo del file `saxon9he.jar` :

```
java -jar "path_to_saxon.jar" -o
petOutput.html -s:pets.xml -xsl:pettransform.xslt
```

Per esempio:

```
java -jar "C:\Program Files\SaxonHE9-7-0-7J\saxon9he.jar" -o
petOutput.html -s:pets.xml -xsl:pettransform.xslt
```

Assicurati di eseguire questo comando su una singola riga.

6. Apri il file di output `petOutput.html` in un editor di testo. Dovrebbe sembrare come questo:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Pets that are available for adoption</title>
  </head>
  <body>

    <h2>Dogs</h2>
    <table id="Dogs">
      <tr>
        <th colname="id">ID</th>
        <th colname="name">Name</th>
        <th colname="vaccinated">Vaccine status</th>
        <th colname="health">Health status</th>
      </tr>
      <tbody>
        <tr>
          <td colname="id">123</td>
          <td colname="name">Sparky</td>
          <td colname="vaccinated">vaccinated</td>
          <td colname="health">healthy</td>
        </tr>
        <tr>
          <td colname="id">234</td>
          <td colname="name">Sadie</td>
          <td colname="vaccinated">unvaccinated</td>
          <td colname="health">sick</td>
        </tr>
        <tr>
          <td colname="id">345</td>
          <td colname="name">Herman</td>
          <td colname="vaccinated">unvaccinated</td>
          <td colname="health">unknown</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

```

    </tbody>
</table>

<h2>Cats</h2>
<table id="Cats">
  <tr>
    <th colname="id">ID</th>
    <th colname="name">Name</th>
    <th colname="vaccinated">Vaccine status</th>
    <th colname="health">Health status</th>
  </tr>
  <tbody>
    <tr>
      <td colname="id">456</td>
      <td colname="name">Cleo</td>
      <td colname="vaccinated">vaccinated</td>
      <td colname="health">healthy</td>
    </tr>
    <tr>
      <td colname="id">567</td>
      <td colname="name">Janet</td>
      <td colname="vaccinated">unvaccinated</td>
      <td colname="health">healthy</td>
    </tr>
    <tr>
      <td colname="id">678</td>
      <td colname="name">Furry</td>
      <td colname="vaccinated">vaccinated</td>
      <td colname="health">sick</td>
    </tr>
  </tbody>
</table>

</body>
</html>

```

7. Apri il file di output `petOutput.html` in un browser web. Dovrebbe mostrare i dati in una tabella semplice.

Leggi Iniziare con xslt online: <https://riptutorial.com/it/xslt/topic/1129/iniziare-con-xslt>

Capitolo 2: Variabili in xslt

Examples

Variabili Xslt

Variabile globale: questa variabile è disponibile ovunque nel foglio di stile xsl. Questa variabile dovrebbe essere solo figlia dell'elemento `<xsl:stylesheet>`.

Variabile locale: questa variabile è disponibile solo dove è dichiarata.

Vedi sotto il codice:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:output omit-xml-declaration="yes"/>

  <xsl:variable name="a" select="5"/>      <!-- Global Variable -->

  <xsl:template match="/">
    <xsl:variable name="b" select="2"/> <!--Local Variable -->
    <xsl:value-of select="$a+$b"/>      <!--Addition of 'a' and 'b' -->
  </xsl:template>
</xsl:stylesheet>
```

L'output del codice sopra sarebbe: **7**

Ci sono due modi per definire un valore per la variabile come:

Con l'espressione xpath nell'attributo **@select dell'elemento <xsl: variable>** come:

```
<xsl:variable name="apple" select="'RED'"/>
```

O

Dal **contenuto dell'elemento <xsl: variable>** come:

```
<xsl:variable name="apple">RED</xsl:variable>
```

Per chiamare una variabile dichiarata usa **\$** Sign con nome variabile, come nel codice sopra '\$ a'

Leggi Variabili in xslt online: <https://riptutorial.com/it/xslt/topic/10060/variabili-in-xslt>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con xslt	Community , Daniel Haley , Dimitre Novatchev , Eero Helenius , JLRishe , Lukasz , Mads Hansen , Mathias Müller , Michael Kay , Nate , Tim McMackin
2	Variabili in xslt	palo , Tim C