

BLOG IN XML

MODELLO XML

```
<news>
  <post>
    <titolo>Yahoo! abbandona MapQuest</titolo>
    <data>08/03/2002</data>
    <testo>
      Yahoo! stringe la cinghia. Il portale che detta le tendenze sull'e-business model, ora rinuncia anche al servizio di cartografia di Aol Time-Warner.
      Il sito di Santa Clara ripiega su un software messo a punto, appositamente, dai propri ricercatori abbandonando il noto programma <link href="http://www.mapquest.com">MapQuest</link>.
      Le pagine per la ricerca dei percorsi stradali e geografici hanno attirato sul portale fondato da Google 11,2 milioni di navigatori.
    </testo>
  </post>
  <post>
    <titolo>Domini: primo passo verso i .kids</titolo>
    <data>07/03/2002</data>
    <testo>
      Gli Stati Uniti avranno un sottodominio dedicato ai più piccoli. L'estensione .kids, in aggiunta al ccTLD (country code Top Level domain) .us, garantirà una navigazione protetta per i minori, assicurando contenuti privi di violenza e pornografia.
    </testo>
  </post>
  <post>
    <titolo>HP-Compaq: nuovo slancio alla fusione</titolo>
    <data>06/03/2002</data>
    <testo>
      A due settimane dal voto del 19 marzo con il quale gli azionisti di
      <link href="http://www.hp.com">Hewlett-Packard</link> decideranno se dare il via alla fusione con Compaq, i sostenitori dell'operazione incassano un insperato incoraggiamento. L'Institutional Shareholder Services (ISS), società specializzata nelle votazioni societarie, ha giudicato la fusione un'ottima mossa nel lungo termine.
    </testo>
  </post>
</news>
```

MODELLO XSL

```
<?xml version="1.0"?>
```

```
<!-- Dichiarazione XML-->
```

```
<!-- Definiamo l'elemento radice. Contiene la dichiarazione della versione e due namespace: quello per XSL e quello per XHTML (senza prefisso)-->
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">
```

```
<!-- Con questa dichiarazione stabiliamo il formato di output. Essendo XHTML dichiariamo anche il DOCTYPE -->
```

```
<xsl:output method="html" indent="yes" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN" />
```

```
<!-- Inizia il foglio di stile. Con questa dichiarazione specifichiamo che applichiamo le egole di trasformazione a partire dall'elemento radice-->
```

```
<xsl:template match="/">
```

```
<!-- Inizia il documento XHTML -->
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<!-- Inizio della testata -->
```

```
<head>
```

```
<title>X-blog</title>
```

```
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
```

```
<!-- Colleghiamo il foglio di stile "folgioStile.css"-->
```

```
<link rel="stylesheet" title="main" href="mainstyle.css" type="text/css" />
```

```
</head>
```

```
<!-- Fine della testata-->
```

```
<!-- Inizia il corpo del documento-->
```

```
<body>
```

```
<!-- Questa parte è identica al documento blog.xhtml". E' la parte fissa del documento-->
```

```
<div id="sezprinc">
```

```
<div id="testata"></div>
```

```
<div id="menu">
```

```
<p><a href="#">home page</a> - <a href="#">about</a> - <a href="#">archivi</a>
```

```
- <a href="#">contatti</a> ----- <a href="#">e-m@il me</a></p>
```

```
</div>
```

```
<div id="contenuto">
```

```
<div id="navigazione">
```

```
<h1>sezione 1</h1>
```

```
<p><a href="#">link 1</a></p>
```

```
<p><a href="#">link 2</a></p>
```

```
<p><a href="#">link 3</a></p>
```

```
<h1>sezione 2</h1>
```

```
<p><a href="#">link 1</a></p>
```

```
<p><a href="#">link 2</a></p>
<p><a href="#">link 3</a></p>
</div>
<!-- Fine della parte fissa del documento-->

<!-- Inizia la sezione "post". Il contenuto è preso dal file "news.xml"-->
<div id="post">

<!-- Selezioniamo l'elemento <post>. Le regole sono applicate in maniera ricorsiva-->
<xsl:for-each select="news/post">

<!-- Ordiniamo i post per data-->
    <xsl:sort data-type="text" select="data" order="descending"/>

<!-- Il testo dell'elemento <data> sarà un titolo <h2>-->
<h1><xsl:value-of select="titolo" /></h1>

<!-- Il testo dell'elemento <titolo> sarà un titolo <h1>-->
<h2><xsl:value-of select="titolo"/></h2>

<!-- Applichiamo il template. Il testo dei post sarà un paragrafo-->
<p><xsl:apply-templates select="testo"/></p>
</xsl:for-each>
</div>
</div>
</div>
</body>

<!-- Fine del corpo del documento XHTML-->
</html>
</xsl:template>

<!-- Dichiariamo il template per la gestione dei link-->
<xsl:template match="link">
    <xsl:element name="a">
        <xsl:attribute name="href">
            <xsl:value-of select="@href"/>
        </xsl:attribute>
        <xsl:apply-templates/>
    </xsl:element>
</xsl:template>

</xsl:stylesheet>
```

FOGLIO DI STILE

/* Stili generici per l'elemento body */

```
body {  
    background : #FFFFFF;  
    color : #FFFFFF;  
    font : 12px Verdana, Geneva, Arial, Helvetica, sans-serif;  
    margin : 0px;  
    padding-bottom : 0px;  
    padding-left : 0px;  
    padding-right : 0px;  
    padding-top : 0px;  
    text-align : center;  
}
```

/* Stile della sezione "sezprinc" */

```
#sezprinc {  
    background: #FFFFFF;  
    border-bottom : 1px solid black;  
    border-left : 1px solid black;  
    border-right : 1px solid black;  
    border-top : 1px solid black;  
    color : #000000;  
    margin : 20px auto;  
    text-align : left;  
    width : 574px;  
}
```

/* Stile della sezione "testata" */

```
#testata {  
    background : #336699;  
    border-bottom : 1px solid black;  
    height : 75px;  
    margin : 0px;  
    padding-bottom : 1px;  
    padding-left : 26px;  
    padding-right : 26px;  
    padding-top : 0px;  
}
```

/* Stile della sezione "contenuto" */

```
#contenuto {  
    border-bottom : #CCCCCC 0px dotted;  
    border-left : #CCCCCC 0px dotted;  
    border-right : #CCCCCC 0px dotted;  
    border-top : #CCCCCC 0px dotted;  
    margin : 0px 27px;  
    padding-bottom : 5px;  
    padding-left : 0px;  
    padding-right : 0px;  
    padding-top : 6px;  
}
```

```
voice-family : inherit;  
width : 512px;
```

```
}
```

/* Stile della sezione navigazione */

```
#navigazione {  
    background : #336699;  
    border-bottom : #CCCCCC 0px;  
    border-left : #CCCCCC 0px;  
    border-right : #CCCCCC 0px dotted;  
    border-top : #CCCCCC 0px;  
    color : #000000;  
    float : left;  
    padding-bottom : 0px;  
    padding-left : 0px;  
    padding-right : 0px;  
    padding-top : 0px;  
    voice-family : inherit;  
    width : 140px;  
}
```

/* Stili dei link della sezione "navigazione" */

```
#navigazione a {  
    background : #336699;  
    color : #99FFFF;  
}
```

```
#navigazione a:hover {  
    background : #336699;  
    color : #DC143C;  
}
```

```
#navigazione a:visited {  
    background : #336699;  
    color : #99FFFF;  
}
```

/* Selettore contestuale. Specifica come appare l'elemento h1 all'interno della sezione "navigazione" */

```
#navigazione h1 {  
    background : #336699;  
    border-bottom : 1px dotted #808080;  
    border-left : 1px dotted #808080;  
    border-right : 1px dotted #808080;  
    border-top : 1px dotted #808080;  
    color : white;  
    font : bold 13px Tahoma, Verdana, Helvetica, sans-serif;  
    margin-bottom : 3px;  
    margin-left : 5px;  
    margin-right : 5px;  
    margin-top : 15px;
```

```
padding-bottom : 2px;  
text-align : center;  
}
```

/* Selettore contestuale. Specifica come appare l'elemento p all'interno della sezione "navigazione" */

```
#navigazione p {  
    background : #336699;  
    color : Black;  
    font : 11px Verdana, Geneva, Arial, Helvetica, sans-serif;  
    margin : 5px 5px;  
}
```

/* Stile della sezione "menu" */

```
#menu {  
    border-bottom : black 1px solid;  
    border-left :black 1px solid;  
    border-right : black 1px solid;  
    border-top : black 0px solid;  
    margin : 0px 26px 1px;  
    padding-bottom : 5px;  
    padding-left : 2px;  
    padding-right : 0px;  
    padding-top : 4px;  
    width : 512px;  
    background : #FFCC00;  
}
```

/* Stili dei link della sezione "menu" */

```
#menu a {  
    background : #FFCC00;  
    color : #CC0000;  
}
```

```
#menu a:visited {  
    background : #FFCC00;  
    color : #CC0000;  
}
```

/* Selettore contestuale. Specifica come appare l'elemento p all'interno della sezione "menu" */

```
#menu p {  
    background : #FFCC00;  
    color : black;  
    font : bold 11px Verdana, Geneva, Arial, Helvetica, sans-serif;  
    text-align : center;  
}
```

/* Stile della sezione "post" */

```
#post {  
    background-color : #FFFFFF;  
    border-bottom : #CCCCCC 0px;  
    border-left : #CCCCCC 0px dotted;
```

```
border-right : #CCCCCC 0px;  
border-top : #CCCCCC 0px;  
color : #000000;  
margin : 0px 0px 0px 142px;  
padding-bottom : 10px;  
padding-left : 16px;  
padding-right : 0px;  
padding-top : 0px;  
voice-family : inherit;  
width : 350px;  
}
```

```
/* Selettore contestuale. Specifica come appare l'elemento h1 all'interno della sezione "post" */  
#post h1 {  
    background : white;  
    color : #191970;  
    font : bold 13px Tahoma, Verdana, Helvetica, sans-serif;  
    margin : 5px;  
}
```

```
/* Selettore contestuale. Specifica come appare l'elemento h2 all'interno della sezione "post" */  
#post h2 {  
    background-color : #FFFFFF;  
    color : #CC0000;  
    font : bold 10px Tahoma, Verdana, Helvetica, sans-serif;  
    margin : 5px;  
}
```

```
/* Selettore contestuale. Specifica come appare l'elemento p all'interno della sezione "post" */  
#post p {  
    background-color : #FFFFFF;  
    border-bottom : 1px dotted black;  
    color : #000000;  
    font : 11px Verdana, Geneva, Arial, Helvetica, sans-serif;  
    line-height : 1.5em;  
    margin : 5px;  
    padding-bottom : 4px;  
}
```

APPUNTI DI XPATH

XPath è un linguaggio XML-based dedicato alla selezione e al filtro di informazioni incluse in un file XML operando sulla struttura del documento XML stesso; XPath fornisce, inoltre, comandi per analizzare, modificare e navigare elementi ed attributi di un documento XML.

XPATH è in sintesi un linguaggio di interrogazione.

La struttura di un percorso XPath è costituita da **step separati dal carattere /** e ogni step è formato da **axis o abbreviated axis**, **axis separator (::)**, un **node test** e da uno o più **node test** e **predicate** (vedi Figura 1).

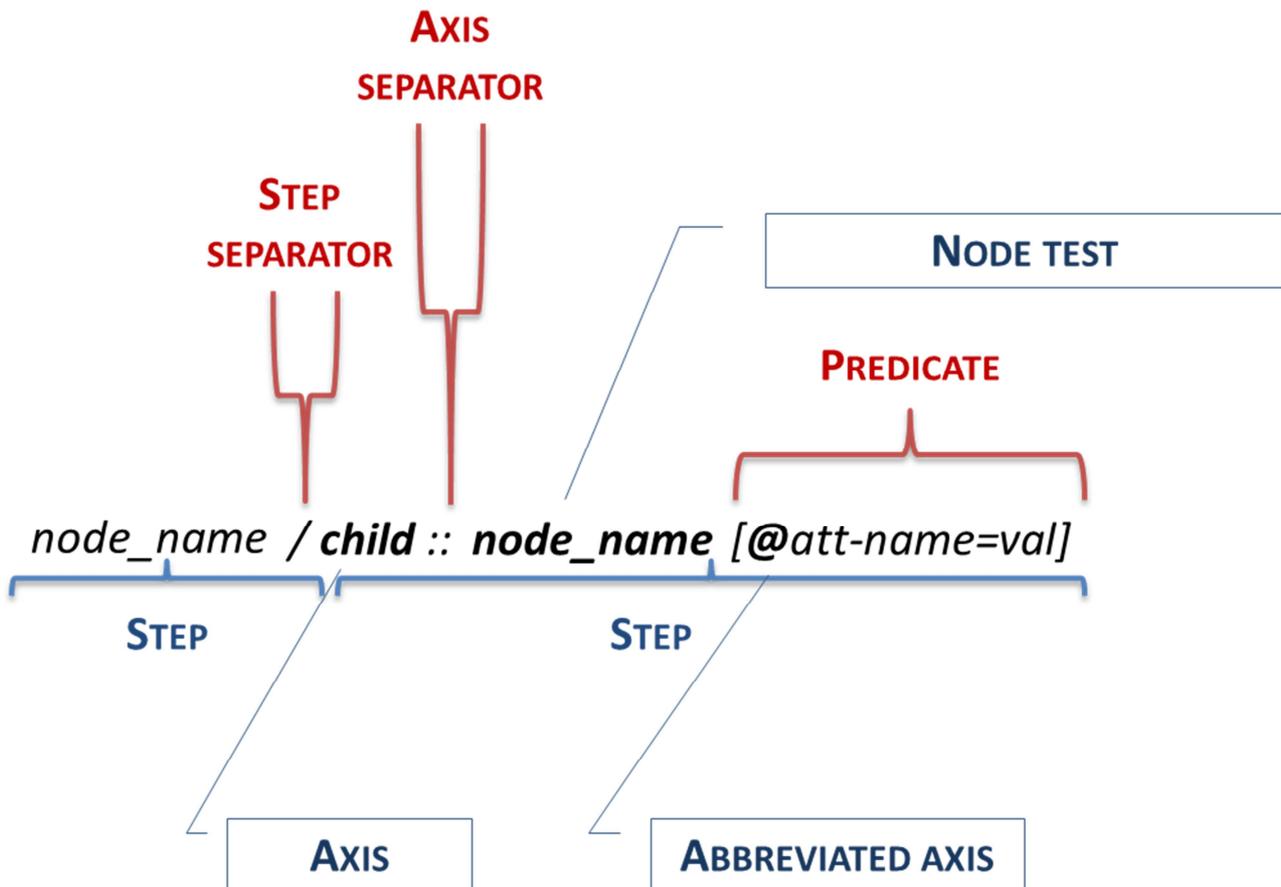


Figura 1 - Schema di un percorso XPath

La formulazione di un'interrogazione con l'utilizzo di XPath richiede la descrizione del percorso a partire dal nodo radice (codificato con *forward slash /*) e l'insieme dei nodi elaborati in ogni *step* è detto **context node**. Analizzando nel dettaglio la struttura rappresentata in Figura 1 si può dire che:

- **axis** : percorso, identificato sull'albero del documento XML, che deve essere utilizzato per la ricerca del **node test** (nell'esempio in Figura 2 *axis child* identifica il percorso formato da i nodi figli del nodo corrente); i valori fondamentali degli axis che possono essere utilizzati in XPath 1.0 sono riassunti nella Tabella 1 e schematizzati in Figura 2;
- **node test**: nome del nodo dell'albero del documento XML che si vuole elaborare (esempio in Figura 2 si intende cercare gli elementi definiti da **tag** con nome uguale al *node test*; *node test notevoli* sono:
 - **comment()**: trova un nodo di commento XML, *per esempio* `<!--Testo del commento -->`
 - **text()**: trova un nodo di tipo testo, *per esempio* `hello in <nodo>hello</nodo>`

- **processing-instruction():** trova istruzioni di processo (**PI – Processing Instructions**) XML, per esempio `<?php echo $a;?>`, per cui in questo caso ('php')
- **node():** trova il nodo.
- **predicate** indica quali sono i criteri con cui filtrare i nodi così identificati (*setNode*) che, nell'esempio di Figura 2 sono quelli che hanno un attributo "**att_name**" istanziato con un valore uguale a **val**.

Valore	setNode risultante
self	context node ¹
parent	nodi "genitori" dei nodi presenti nel context node
child	nodi figli dei nodi presenti nel context node
descendant	nodi discendenti dei nodi presenti nel context node esclusi i nodi attributo
descendant-or-self	stesso setNode di <i>descendant</i> più il context node stesso
ancestor	nodi antenati dei nodi presenti nel context node fino ad arrivare al nodo root
ancestor-or-self	stesso setNode di <i>ancestor</i> più il context node stesso
attribute	nodi attributo del context node
following	nodi discendenti, secondo l'ordine definito nel documento XML, dai nodi del context node, esclusi i nodi attributo
following-sibling	nodi "fratelli" (medesimo livello), secondo l'ordine definito nel documento XML, dei nodi del context node, esclusi i nodi attributo
preceding	nodi che precedono, secondo l'ordine definito nel documento XML, i nodi del context node, esclusi i nodi attributo
preceding-sibling	nodi "fratelli" (medesimo livello), secondo l'ordine definito nel documento XML, che precedono i nodi del context node, esclusi i nodi attributo

OPERATORE	DESCRIZIONE	OPERATORE	DESCRIZIONE
=	uguaglianza	+	addizione
!=	disuguaglianza	-	sottrazione
> (>)	maggiore	*	moltiplicazione
< (<)	minore	div	divisione
>=	maggiore uguale	mod	resto di una divisione
<=	minore uguale	round()	arrotonda all'intero più vicino
and	AND logico	ceiling()	arrotonda per difetto all'intero più vicino (soffitto)
or	OR logico	floor()	arrotonda per eccesso all'intero più vicino
not()	NOT, negazione logica	sum()	somma un insieme di numeri interi
position()	posizione del nodo		unisce 2 setNode
OPERATORE DI STRINGHE	DESCRIZIONE	OPERATORE DI STRINGHE	DESCRIZIONE
string(object?)	converte i 4 tipi di dati ² di XPath in una stringa	contains(s1, s2)	restituisce true se la stringa s1 contiene la stringa s2
concat(string, string, string*)	concatena le stringhe	normalize-space(string?)	tutti i <i>whitespace</i> all'inizio e alla fine della stringa vengono rimossi e tutte le sequenze di <i>whitespace</i> sono sostituite da un singolo spazio

¹ Per **context node** si intende il **nodo corrente**, ovvero il nodo da cui si considerano descritti i path nel contesto della query; all'inizio di una query XPath il context node costituito solo dal nodo radice (root).

² I tipi di nodi di XPath sono: node-set (insiemi di nodi privi di un ordine intrinseco), stringhe, numeri e booleani.
XML, XSL e CSS esercizio BLOG (6 maggio 2011)
Note di XPath

Di seguito si riportano alcune strutture standard generalizzate per la selezione di nodi di un albero di un documento XML:

1. `/child::node_A/child::node_B` restituisce un **setNode** costituito da tutti i nodi dell'albero XML con nome (*tag*) **node_B** che discendono del nodo root (*/*) e che hanno un padre con nome (*tag*) **node_A**.
2. `/descendant::node_A` restituisce un **setNode** costituito da tutti i nodi dell'albero XML con nome **node_A discendenti (descendant)** del nodo *root* (*/*).; la differenza con la *query* precedente è che il **setNode** restituito in questo caso è quello formato da tutti i nodi dell'albero XML con nome (*tag*) **node_A** che vengono ricercati in tutto il documento XML dell'albero e non solo come figli di un determinato nodo.
3. `/descendant::node_A[attribute::att_A='valA']` restituisce un **setNode** costituito da tutti i nodi etichettati con **node_A** che hanno un attributo di nome **att_A** istanziato con il valore **valA**
4. `/descendant::node_A/child::*` restituisce un **setNode** costituito da tutti i nodi figli dell'elemento **node_A**
5. `/descendant::*` restituisce un **setNode** costituito da tutti i nodi del documento XML

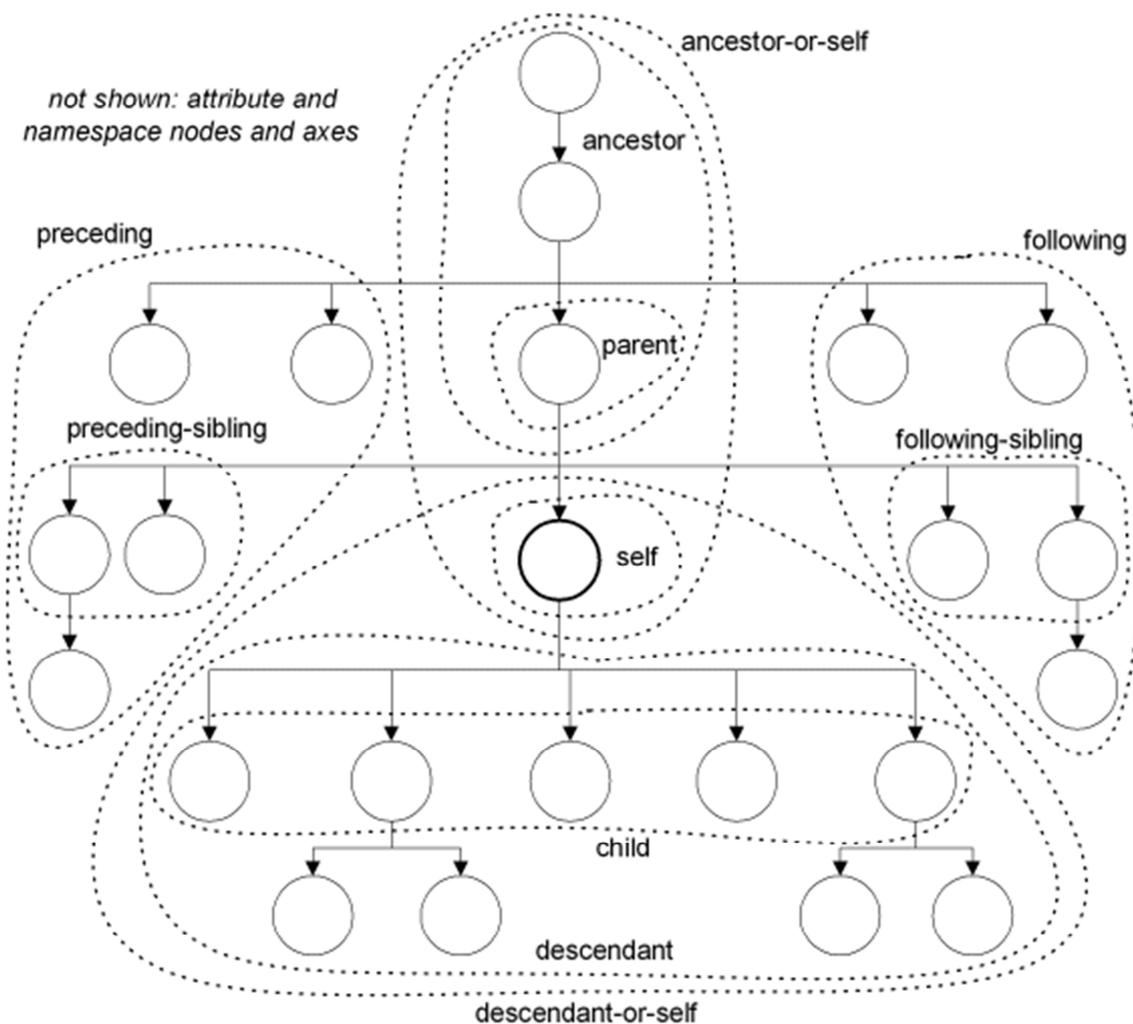


Figura 2 - XPath Axes (immagine tratta da Norman Walsh, <http://nwalsh.com/docs/tutorials/xsl/xsl/foil22.html>)