

XQuery

XQuery (*XML Query Language*) è il linguaggio che consente di fare manipolazioni e interrogazioni (*query*) su un documento XML, in modo analogo al linguaggio SQL con il quale si costruiscono query per i database.

È uno standard del **W3C** (*World Wide Web Consortium*), l'Ente internazionale che regola il Web. XQuery è un linguaggio in fase di sviluppo.

Le specifiche del linguaggio (versione XQuery 1.0, 2007) si trovano all'indirizzo:

<http://www.w3.org/TR/xquery/>

La versione più aggiornata, ma non definitiva (XQuery 3.0, gennaio 2013) si trova all'indirizzo:

<http://www.w3.org/TR/xquery-30/>

I file contenenti codice XQuery hanno di solito estensione **.xq** oppure **.xquery**.

Per poter eseguire le interrogazioni XQuery su un documento XML si devono usare interpreti del linguaggio, detti anche motori o processori (*XQuery engine* o *XQuery processor*), per esempio i software *Saxon* o *XQilla*.

Di seguito viene presentato un esempio di uso del linguaggio XQuery con il motore *Saxon*.

Il software Saxon viene scaricato dal sito

<http://saxon.sourceforge.net>

In particolare la versione HE è *open source* e di uso libero.

Il download rende disponibile il file eseguibile Java **saxon9he.jar**, che può essere utilizzato da linea comandi.

Si può aprire il *Prompt dei comandi* dalla cartella contenente il file scaricato e scrivere il comando:

```
c:\Programmi\java\jre7\bin\java.exe -cp saxon9he.jar net.sf.saxon.Query %*
```

Per aprire una finestra di Prompt dei comandi da una cartella in Windows7/Windows8

1. Selezionare la cartella che si desidera aprire nel *Prompt dei comandi*
2. Fare clic con il tasto destro su di essa tenendo premuto il tasto *Shift* (Maiuscolo)
3. Scegliere **Apri finestra di comando qui**.

Si può anche creare un file *batch* e scrivere il suo nome nel *Prompt dei comandi*.

File **saxon.bat**

```
@echo off  
c:\Programmi\java\jre7\bin\java.exe -cp saxon9he.jar net.sf.saxon.Query %*
```

Utilizziamo un semplice documento XML che descrive le informazioni dei libri di una biblioteca.

(biblioteca.xml)

```
<?xml version="1.0" ?>
<biblioteca>
<libro>
<autore>Manzoni Alessandro</autore>
<titolo>I promessi sposi</titolo>
<annoedizione>1840</annoedizione>
</libro>
<libro>
<autore>William Shakespeare</autore>
<titolo>Hamlet</titolo>
<annoedizione>1654</annoedizione>
</libro>
</biblioteca>
```

Prepariamo ora l'interrogazione per estrarre dall'archivio dei libri l'elenco con i soli titoli: è un elenco puntato realizzato con i tag HTML ` ... `.

La sintassi del comando XQuery utilizza una ripetizione **for** su tutti i libri dell'elenco. La funzione **doc()**, avente come argomento il nome del file XML, è l'inizio di un'espressione **XPath**:

```
doc("biblioteca.xml")/biblioteca/libro
```

L'elenco viene ordinato con la clausola **order**. Per indicare i dati che devono essere visualizzati, come risultato dell'interrogazione, si usa la parola chiave **return**. Si osservi che l'output contiene tag HTML e i dati contenuti negli item del documento XML, ottenuti con la funzione **data**, racchiusa tra una coppia di parentesi graffe.

File dell'interrogazione XQuery (elencotitoli.xq)

```
<ul>
{
for $x in doc("biblioteca.xml")/biblioteca/libro
order by $x/titolo
return <li>{data($x/titolo)}</li>
}
</ul>
```

Utilizzando il comando *batch* **saxon.bat**, dal *Prompt dei comandi*, scriviamo:

```
saxon elencotitoli.xq
```

Esso produce a video l'elenco dei titoli.

Aggiungendo l'opzione **-o** si può indicare, dopo i due punti, il nome di un file come output dell'interrogazione, per esempio un file *titoli.html*, che diventa una pagina Web visualizzabile con un browser, oppure un file *titoli.xml* che diventa a sua volta un documento XML.

```
saxon elencotitoli.xq -o:titoli.html
```

Un secondo esempio mostra il file `.xq` completo di un'intestazione e dei tag di una pagina Web, con la visualizzazione dell'anno di edizione insieme al titolo, separati da un punto.

File dell'interrogazione XQuery (titolianni.xq)

```
<html>
<body>
<h1>Biblioteca</h1>
<ul>
{
for $x in doc("biblioteca.xml")/biblioteca/libro
order by $x/titolo
return
<li>{data($x/titolo)}. Anno: {data($x/annoedizione)}</li>
}
</ul>
</body>
</html>
```

L'esecuzione dell'interrogazione si ottiene con un comando come il seguente:

```
saxon titolianni.xq -o:elencolibri.html
```

Nelle interrogazioni XQuery si possono inserire anche criteri di selezione attraverso la clausola **where**.

Per esempio, la seguente interrogazione estrae i titoli dei libri aventi l'anno di edizione superiore al 1800:

```
where $x/annoedizione>1800
```

File dell'interrogazione XQuery (titolidopo1800.xq)

```
<ul>
{
for $x in doc("biblioteca.xml")/biblioteca/libro
where $x/annoedizione>1800
order by $x/titolo
return <li>{data($x/titolo)}</li>
}
</ul>
```

Eseguendo l'interrogazione con il comando:

```
saxon titolidopo1800.xq -o:elenco1800.html
```

si ottiene un elenco puntato con i soli titoli dei libri pubblicato dopo il 1800.

Nel codice XQuery si possono inserire anche espressioni di assegnazione ad una variabile con la clausola **let**:

```
let $a := 1
```

Per esempio, la seguente interrogazione produce in output i titoli dei libri con il numero di anni trascorsi dalla pubblicazione, indicando con una variabile l'anno attuale:

File dell'interrogazione XQuery (annidistanza.xq)

```
<ul>
{
for $x in doc("biblioteca.xml")/biblioteca/libro
let $attuale := 2013
order by $x/titolo
return
<li>{data($x/titolo)}. {$attuale - data($x/annoedizione)} anni fa </li>
}
</ul>
```

La struttura generale di un'interrogazione XQuery è quindi:

```
for ...
let ...
where ...
order by ...
return ...
```

Le clausole non sono obbligatoriamente tutte presenti in una stessa interrogazione. La struttura si chiama di solito **espressione FLWOR** dalle iniziali delle parole chiave (si legge *flower*).