



**Sample XML:**

**let $bookstore:=**

<bookstore>

<book category=”none”>Book yet to be published.</book>

<book category="COOKING">Cooking Books

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="CHILDREN">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="WEB">

<title lang="en">XQuery Kick Start</title>

<author>James McGovern</author>

<author>Per Bothner</author>

<author>Kurt Cagle</author>

<author>James Linn</author>

<author>Vaidyanathan Nagarajan</author>

<year>2003</year>

<price>49.99</price>

</book>

<book category="WEB">XML Books

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

<book category="Programming Language">

<title lang="en">Core Java</title>

<author>H.Shieldit</author>

<year>2012</year>

<price>100</price>

</book>

<book category="Programming Language">

<title lang="en">C</title>

<author>Y.Kanetkar</author>

<year>2012</year>

<price>50</price>

</book>

<book category="Programming Language">

<title lang="en">C++</title>

<author>Y.Kanetkar</author>

<year>2013</year>

<price>60</price>

</book>

<book category="Programming Language">

<title lang="en">C#</title>

<author>Wrox</author>

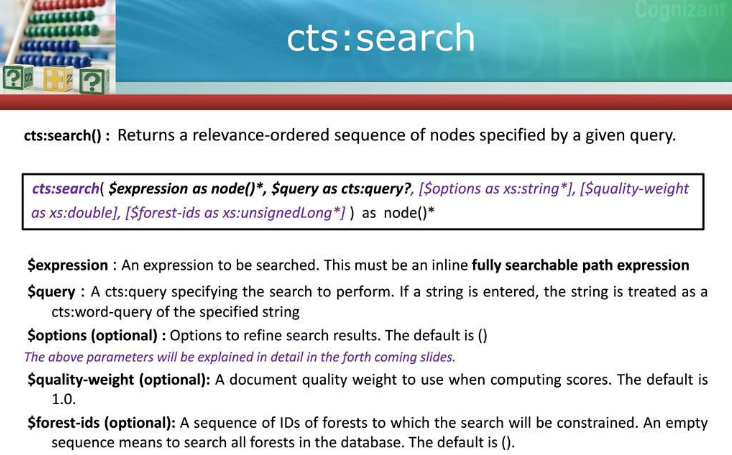
<year>2013</year>

<price>110</price>

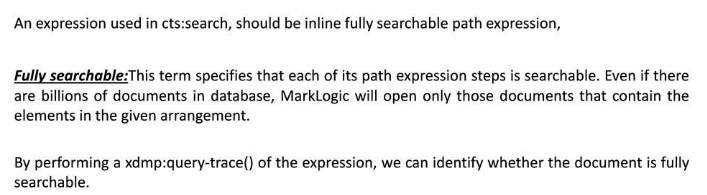
</book>

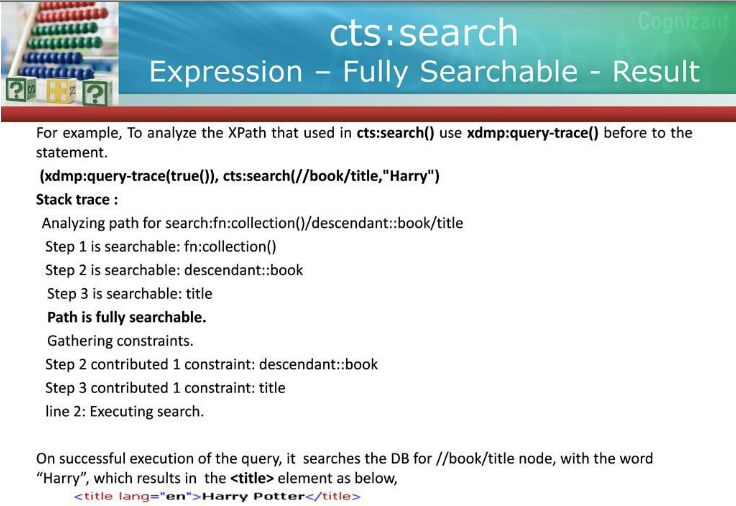
</bookstore>

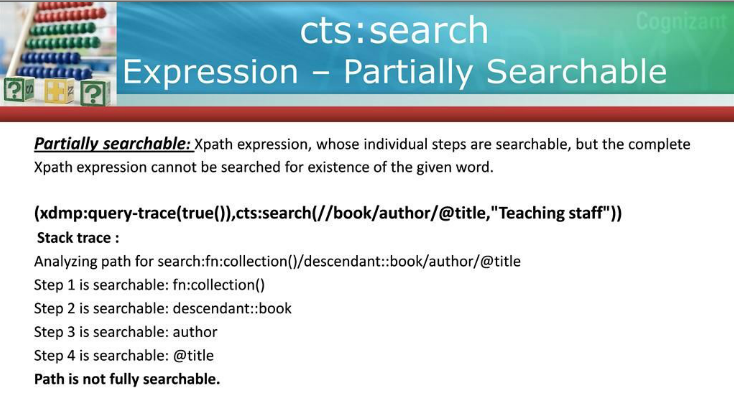
**return xdmp:document-insert("/books.xml",$bookstore,(),"/online")**

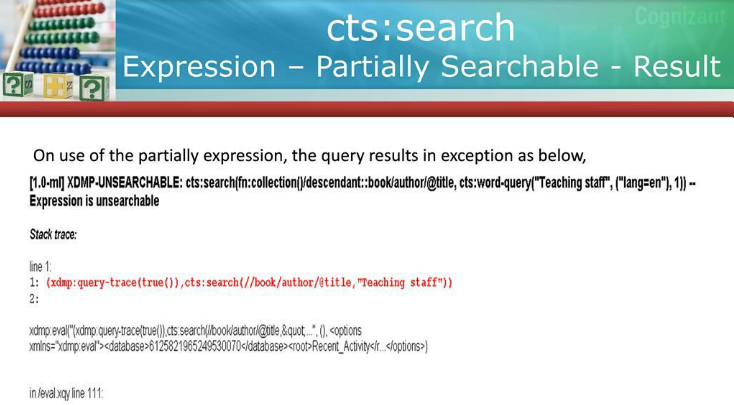
****

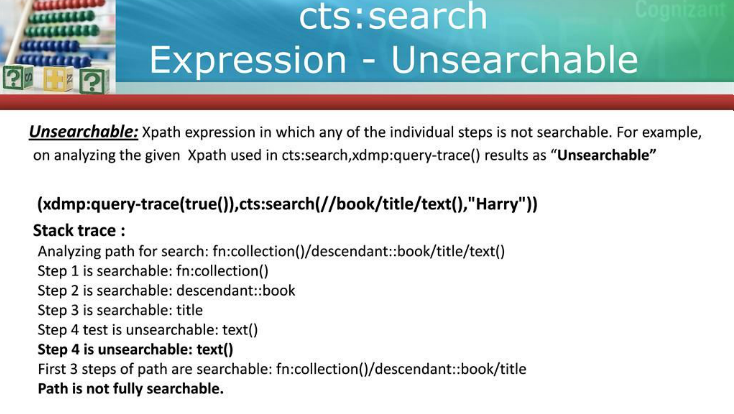
**CTS: SEARCH- Expression should be fully searchable:**

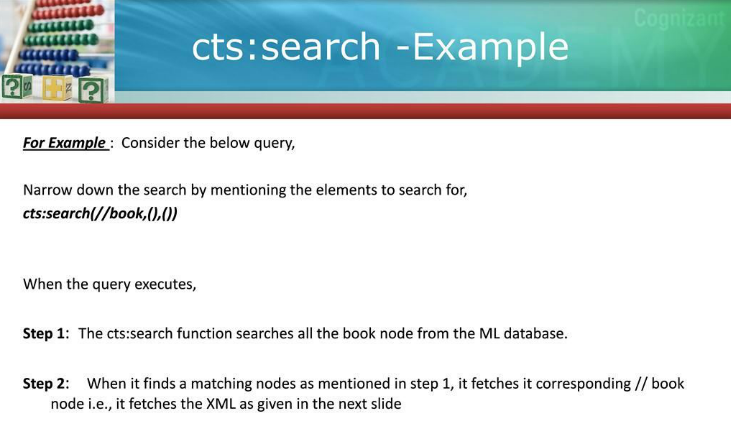
****

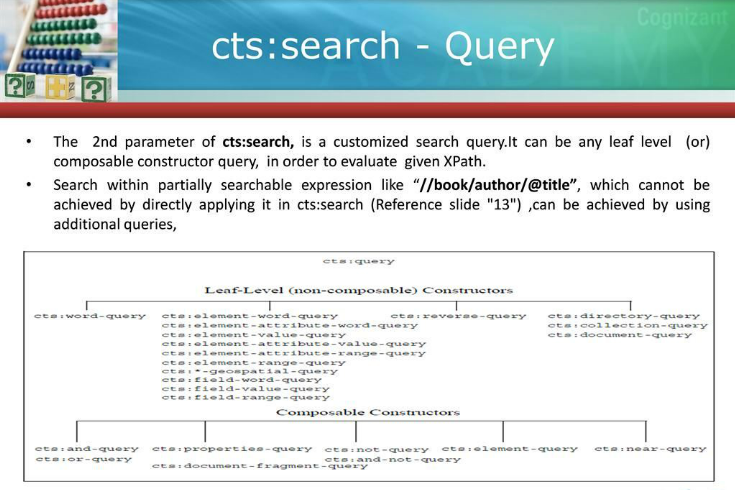
****

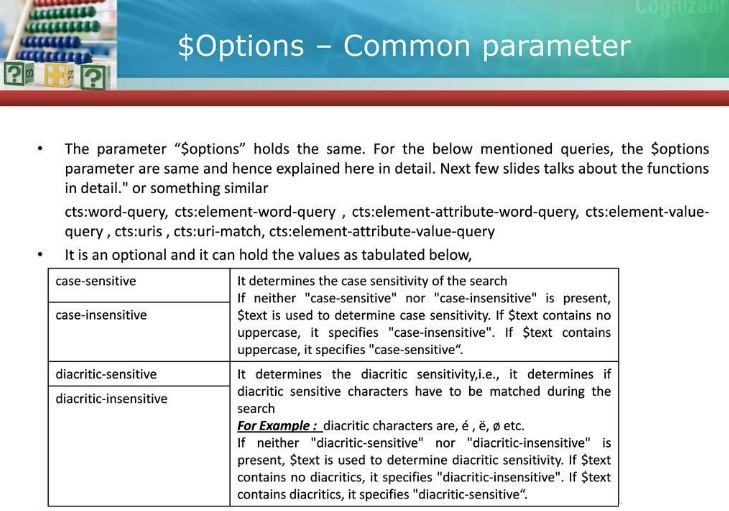
****

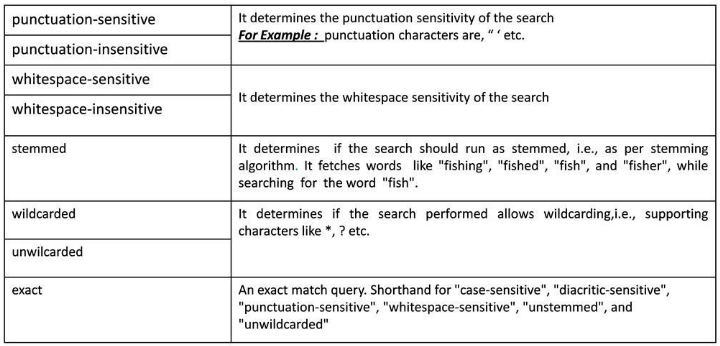


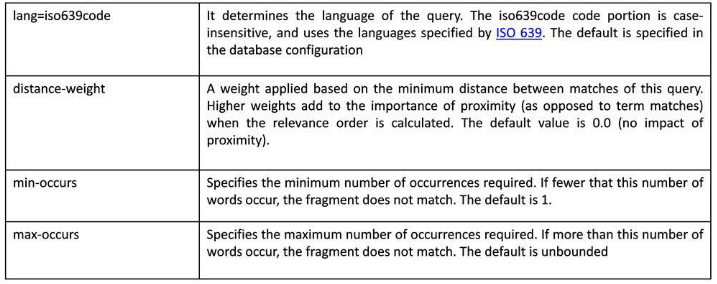


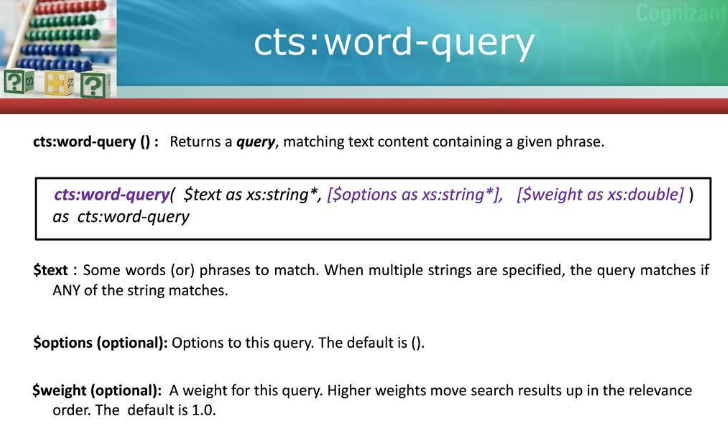


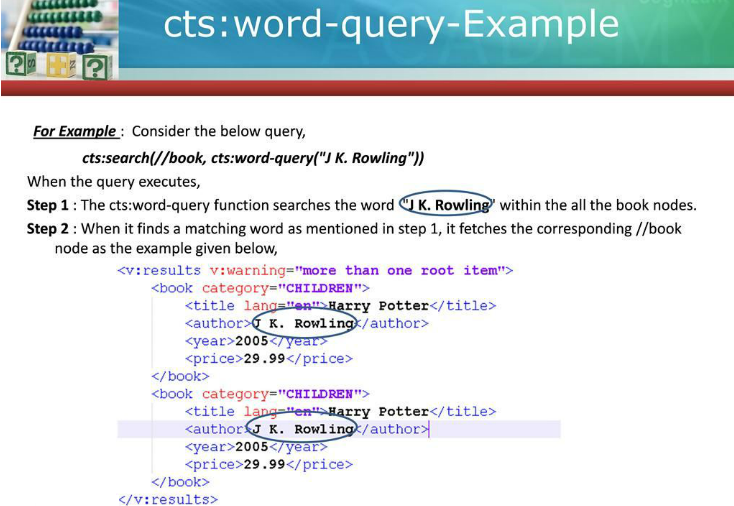


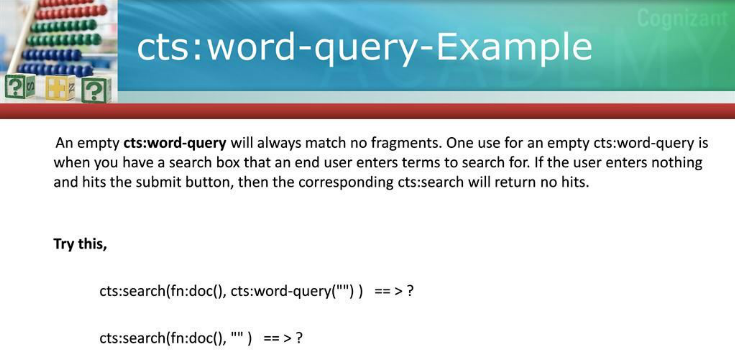


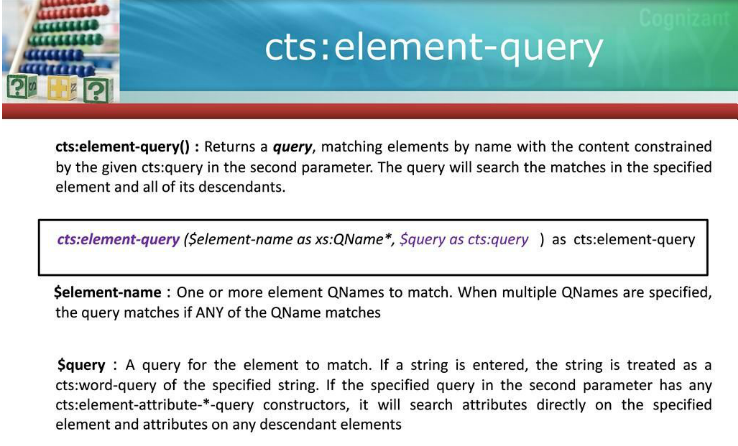












**Example1:**

***cts:search(fn:collection()/bookstore/book,cts:element-query((xs:QName("title"),xs:QName("author")),cts:word-query(("Italian","Rowling"))))***

* Search is performed on the descendants of **book** element on title and author element values.

**<results warning="more than one root item">**

<book category="COOKING">Cooking Books

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="CHILDREN">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

**</results>**

**Example2:**

***cts:search(//book,cts:element-query(xs:QName("book"),"39.95"))***

* *<book category="WEB">XML Books*

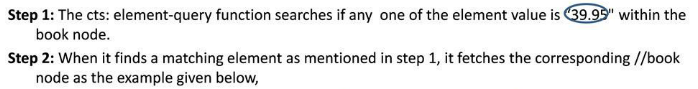
*<title lang="en">Learning XML</title>*

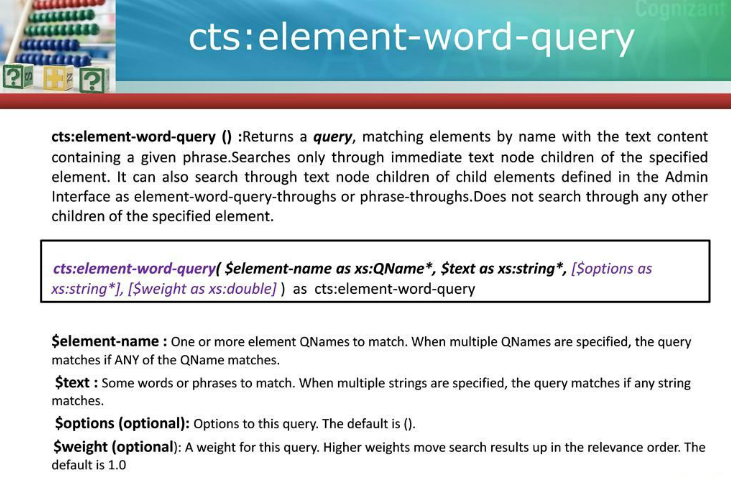
*<author>Erik T. Ray</author>*

*<year>2003</year>*

*<price>39.95</price>*

*</book>*





**Example1:**

The cts:element-word-query searches for “XML” keyword in the every immediate text node children of **title** element in **book** element.

***cts:search(/bookstore/book,cts:element-word-query(xs:QName("title"),"XML"))***

* ***Result:***

*<book category="WEB">XML Books*

*<title lang="en">Learning XML</title>*

*<author>Erik T. Ray</author>*

*<year>2003</year>*

*<price>39.95</price>*

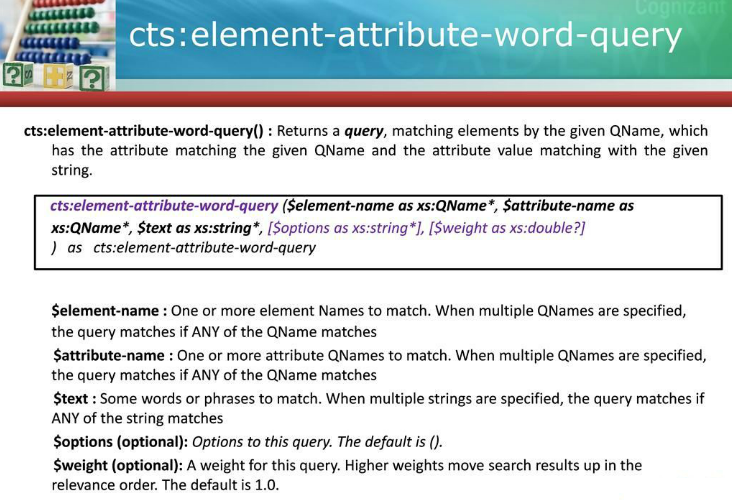
*</book>*

**Example2:**

***cts:search(/bookstore/book,cts:element-word-query(xs:QName("book"),"XML"))***

* ***Result:***

Empty sequence i.e. no result, because book element does not have immediate text node children.



**Example1:**

This will search for all matching “**book**” elements where attribute name is “**category**” with value “**Language**”.

***cts:search(/bookstore/book,cts:element-attribute-word-query(xs:QName("book"),xs:QName("category"),"Language"))***

* **Result**

<results warning="more than one root item">

<book category="Programming Language">

<title lang="en">Core Java</title>

<author>H.Shieldit</author>

<year>2012</year>

<price>100</price>

</book>

<book category="Programming Language">

<title lang="en">C</title>

<author>Y.Kanetkar</author>

<year>2012</year>

<price>50</price>

</book>

<book category="Programming Language">

<title lang="en">C++</title>

<author>Y.Kanetkar</author>

<year>2013</year>

<price>60</price>

</book>

<book category="Programming Language">

<title lang="en">C#</title>

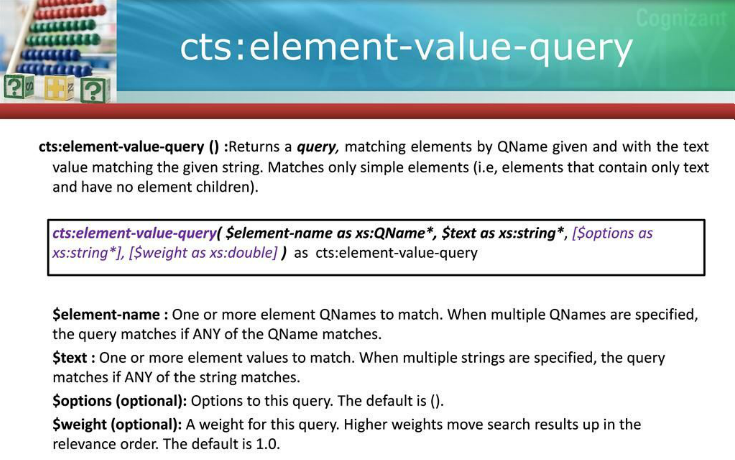
<author>Wrox</author>

<year>2013</year>

<price>110</price>

</book>

</results>



**Note:** cts:element-value-query(..) always looks for exact match i.e. match will be found only when exact keyword is searched for element value.

**Example1:**

***cts:search(/bookstore/book,cts:element-value-query(xs:QName("book"),"Cooking Books"))***

**Result:**

***->***

Empty sequence i.e. no result, because book element is not Simple element, it is a complex element.

**Example2:** Search for exact value only i.e. if text is the exact keyword for an element value.

**cts:search(//book,cts:element-value-query(xs:QName("book"),"Book yet to be published.")) Result:**

***->*** *<book>Book yet to be published.</book>*

**Example3:** Search for exact value only i.e. if text is the exact keyword for an element value.

**cts:search(//book,cts:element-value-query(xs:QName("book")," published.")) Result:**

***->***

Empty sequence i.e. no result, because book element does not have exact keyword as “published”.

**Difference between cts:element-word-query(..) and cts:element-value-query(..):  
cts:element-word-query(..) :**

It will return result by searching the immediate text node of **title** element **using contains based search**.

cts:search(//book,cts:element-word-query(xs:QName("title")," Italian"))

Result:

<book category="COOKING">Cooking Books

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

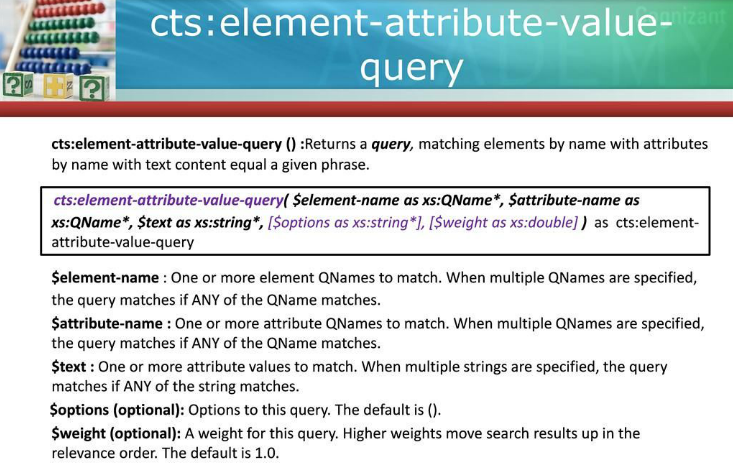
**cts:element-value-query(..) :**

It will return result by searching the immediate text node of **title** element **using exact value match based search**.

cts:search(//book,cts:element-value-query(xs:QName("title")," Italian"))

Result:

Empty sequence i.e. “Italian” is not present as an exact value of title element in the xml/database.



**Note:** cts:element-attribute-value-query(..) always looks for exact match i.e. match will be found only when exact keyword is searched for attribute value of an element.

**Example1:**

***cts:search(//book,cts:element-attribute-value-query(xs:QName("book"),xs:QName("category"),"Programming Language"))***

**Result:**

<results warning="more than one root item">

<book category="Programming Language">

<title lang="en">Core Java</title>

<author>H.Shieldit</author>

<year>2012</year>

<price>100</price>

</book>

<book category="Programming Language">

<title lang="en">C</title>

<author>Y.Kanetkar</author>

<year>2012</year>

<price>50</price>

</book>

<book category="Programming Language">

<title lang="en">C++</title>

<author>Y.Kanetkar</author>

<year>2013</year>

<price>60</price>

</book>

<book category="Programming Language">

<title lang="en">C#</title>

<author>Wrox</author>

<year>2013</year>

<price>110</price>

</book>

</results>

**Example1:**

***cts:search(//book,cts:element-attribute-value-query(xs:QName("book"),xs:QName("category")," Language"))***

**Result:**

Empty sequence i.e. no result, because ***category*** attribute does not have exact keyword as “Language” in the xml doc/database.

**Difference between cts:element-attribute-word-query(..) and cts:element-attribute-value-query(..):**

